



**Universidad Católica
"Nuestra Señora de la Asunción"
Campus Itapúa**

Facultad de Ciencias y Tecnología

Ingeniería Informática

Tesis de Grado

**Previsión meteorológica del tiempo para
Encarnación en un clúster de Raspberry
Pi's utilizando el Modelo de Investigación y
Pronóstico del Tiempo (WRF)**

Año 2016



**Universidad Católica
“Nuestra Señora de la Asunción”
Campus Itapúa**

Facultad de Ciencias y Tecnología

Ingeniería Informática

Tesis de Grado

**Previsión meteorológica del tiempo para Encarnación
en un clúster de Raspberry Pi's utilizando el Modelo
de Investigación y Pronóstico del Tiempo (WRF)**

**Ever Daniel Barreto Rojas
Nicolás Rodríguez Müller**

**Tutora de Tesis
Ing. Aura Rocío Villalba Alderete**

**Cotutor de Tesis
MSc. Lic. Julián Báez**

Año 2016

Dedicatoria

*A mi hija Maia Anahí, a mi hijo Santiago
Daniel y a mi esposa Miryam*

Ever Daniel Barreto Rojas

A mi familia

Nicolás Rodríguez Müller

Agradecimientos

*Los tesistas queremos expresar un profundo agradecimiento
a nuestra tutora, la Ing. Aura Villalba,
y a nuestro cotutor el MSc. Lic. Julián Báez Benítez
por su constante apoyo y ayuda
para la realización de este trabajo,
al Ing. Julián Cáceres por su
incondicional ayuda en tiempos difíciles,
al Físico Fernando Pío Barrios y al Ing. Wilson Caballero
por su desinteresada ayuda, paciencia y orientación
durante el desarrollo de esta investigación.
Muchas Gracias*

*También queremos agradecer a todo el equipo humano de la
Dirección de Meteorología e Hidrología
por todo el apoyo que nos brindaron
y por su importante labor de monitoreo buscando
la protección de la vida humana y la propiedad
Muchas Gracias*

*Agradezco a mi familia,
a mi papá Juan de Dios,
a mi mamá Josefina Leonor
y a mi hermano Hugo David
por su incondicional e incansable
apoyo, soporte y paciencia
durante todos estos años.*

*A mi familia política,
a Don Herculano y a Doña Blanca
por su apoyo y comprensión
en éstos meses de intenso trabajo*

Ever Daniel Barreto Rojas.

*Especial agradecimiento
a mi familia por toda la comprensión y
ayuda brindada en el transcurso
del presente proyecto y siempre.
Muchas Gracias*

Nicolás Rodríguez Müller

Índice de Contenido

Dedicatoria	3
Agradecimientos.....	4
Índice de Contenido.....	6
Índice de Ilustraciones.....	11
Índice de Tablas.....	15
Resumen	16
Abstract.....	17
1 Introducción	18
1.1 Definición del tema.....	18
1.2 Planteamiento general.....	18
1.3 Problemas a resolver.....	19
1.4 Fundamentación.....	19
1.5 Objetivo general.....	19
1.6 Objetivos específicos	19
1.7 Alcance	20
1.8 Contenido del libro	20
2 Meteorología	22
2.1 Orígenes	22
2.2 La época instrumental y el desarrollo de la navegación	22
2.3 La institucionalización de la meteorología en el siglo XIX.....	24
2.4 La meteorología moderna	24
2.5 Tiempo y Clima	27
2.6 La Atmósfera terrestre	27
2.7 Capas de la Atmósfera	27
2.8 Vapor de Agua	29
2.9 Meteoros	29
2.10 Escalas Meteorológicas.....	30

3	Predicción Numérica del Tiempo.....	31
3.1	Breve introducción.....	31
3.2	Historia.....	32
3.3	Proceso de inicialización	33
3.4	Modelos	33
3.5	Parametrización	34
3.6	Dominios.....	35
4	Modelo de Investigación y Pronóstico del Tiempo (WRF)	37
4.1	Reseña Histórica	37
4.2	Características principales y utilización	37
4.3	Motores disponibles.....	38
4.4	Condiciones iniciales	38
4.4.1	El Modelo GFS.....	39
4.5	Componentes	39
5	Clústeres de Computadoras.....	41
5.1	Breve Introducción	41
5.2	Beneficios de la tecnología clúster	42
5.3	Clasificación de los clústeres.....	42
5.4	Componentes de un Clúster	43
5.4.1	Nodos.....	43
5.4.2	Almacenamiento.....	43
5.4.3	Conexiones de Red.....	44
5.4.4	Middleware.....	44
5.4.5	Ambientes de Programación Paralela.....	45
6	Hardware Utilizado	46
6.1	Estaciones Meteorológicas Automáticas	46
6.1.1	Ventajas	46
6.1.2	Desventajas.....	47
6.1.3	Sensores.....	47
6.1.4	Almacenamiento de datos.....	48
6.1.5	Transmisión de Datos	48
6.2	Estaciones Hidrológicas Automáticas	49
6.2.1	Sensores.....	49
6.3	Tarjetas de Memoria SD	50
6.3.1	Formatos de Tarjeta.....	50
6.3.2	Formato Secure Digital.....	50
6.4	Computadoras de Placa Reducida.....	51
6.4.1	Tipos de SBC.....	51
6.4.2	Elección de un SBC.....	51

6.4.3	Selección del SBC para el proyecto	52
6.4.4	Raspberry Pi	52
6.5	Clúster de Raspberry Pi's	54
6.5.1	Componentes principales del clúster	55
6.5.2	Constitución del clúster	55
6.6	Servidor de Pruebas	59
7	Software utilizado	60
7.1	Sistemas Operativos.....	60
7.1.1	Elección del Sistema Operativo.....	60
7.2	Lenguajes de programación	61
7.2.1	PHP.....	61
7.2.2	JavaScript	62
7.2.3	Python.....	62
7.2.4	NCL	62
7.3	Bases de Datos	63
7.3.1	MySQL	63
7.3.2	PostgreSQL.....	63
7.4	Cola de Mensajes	63
7.4.1	Beanstalk	63
7.5	Frameworks	64
7.5.1	Laravel.....	64
7.5.2	Vue.js.....	64
7.5.3	AngularJS	64
7.5.4	Ionic	65
7.6	Librerías	65
7.6.1	Laravel Collective SSH	65
7.6.2	Pheanstalk.....	65
7.6.3	Guzzle.....	65
7.6.4	NumPy.....	65
7.6.5	netCDF4 module	66
7.6.6	MPICH	66
7.7	Utilitarios	66
7.7.1	Win32DiskImager	66
7.7.2	Ansible.....	66
8	Configuraciones, parametrizaciones, ajustes y validación del Modelo WRF.....	68
8.1	Compilación del Modelo WRF para la Arquitectura ARM.....	68
8.1.1	Preparación de Imágenes para los Nodos del Clúster	69
8.2	Configuración del dominio	69
8.3	Parametrización del Dominio	70
8.4	Configuración del WRF en el Clúster.....	71
8.5	Validación del WRF	72
9	Sistemas desarrollados	73

9.1	GEMHA.....	73
9.1.1	Antecedentes y análisis de la situación actual.....	73
9.1.2	Detalles del sistema	74
9.1.3	Componentes del sistema	76
9.2	WRF Runner	81
9.2.1	Proceso de ejecución del modelo WRF.....	81
9.2.2	Componentes del sistema	82
9.3	Aplicación Móvil	86
9.3.1	Análisis	87
9.3.2	Detalles del sistema	87
10	Pruebas realizadas	89
10.1	GEMHA.....	89
10.1.1	Proceso de datos de estaciones.....	89
10.2	Compilación del WRF en el Raspberry Pi.....	90
10.2.1	Compilación y prueba del MPI.....	90
10.2.2	Compilación del WRF.....	91
10.3	Configuración del clúster para el WRF	94
10.3.1	Configuración del NFS.....	94
10.3.2	Configuración de los clientes NFS	94
10.3.3	Verificación de ejecución del modelo	95
10.3.4	Tiempos de ejecución	96
10.4	Parametrización del WRF.....	97
10.4.1	Parametrización del WRF.....	97
10.5	WRF Runner	100
10.5.1	Pruebas de conexión.....	101
10.6	Validación de pronósticos del WRF	101
10.6.1	Análisis de los archivos GFS.....	101
10.6.2	Ejecución de las Simulaciones	102
10.6.3	Comparación de los datos.....	104
10.6.4	Resultados.....	104
11	Conclusiones y trabajos futuros.....	110
11.1	Conclusiones.....	110
11.2	Trabajos futuros.....	111
12	Anexo	112
12.1	Red de estaciones de la DMH.....	112
12.1.1	Estaciones Meteorológicas Automáticas.....	112
12.1.2	Estaciones Hidrológicas Automáticas	113
12.1.3	Datos en bruto de a Setiembre 2016.....	113
12.2	Costos del proyecto.....	115
12.3	Montaje del clúster de RPi's.....	115
12.4	Carencia de un modelo de predicción numérica para Encarnación	119

12.5	Validación del modelo WRF	120
12.6	Presentación Centro de monitoreo y alerta hidrometeorológica de Encarnación	121
12.7	Taller de Hidrometeorología Aplicada	122
12.8	Visita técnica a la DMH.....	124
12.9	Montaje de Rack con servidor de prueba en el Datacenter de la DMH.....	125
13	Glosario de términos.....	127
14	Bibliografía.....	129

Índice de Ilustraciones

Ilustración 1: Mapa de depresión sobre el este de Estados Unidos en 1842. Fuente: Elias Loomis.....	23
Ilustración 2: Globo sonda. Fuente: Servicio Meteorológico de los Estados Unidos ..	25
Ilustración 3: Atmósfera terrestre. Fuente: Taller de Hidrometeorología UCI - 2016	27
Ilustración 5: Capas de la atmósfera. Fuente: Taller de Hidrometeorología UCI - 2016	28
Ilustración 4: Variación térmica en la atmósfera. Fuente: Taller de Hidrometeorología UCI - 2016	28
Ilustración 6: Vapor de agua en la atmósfera. Fuente: Taller de Hidrometeorología UCI - 2016	29
Ilustración 7: Eniac. Fuente: U. S. Army Photo.	32
Ilustración 8: Dominio del modelo. Fuente: The Comet Program.	36
Ilustración 9: Componentes del WRF. Fuente: Elaboración propia.....	40
Ilustración 10: Tamaños de Tarjeas SD. Fuente: SD Association.....	50
Ilustración 11: Raspberry Pi 3 Modelo B. Fuente: Raspberry Pi Foundation.	53
Ilustración 12: Componentes internos del Raspberry Pi. Fuente: Raspberry Pi Foundation.	53
Ilustración 13: Esquema Conceptual del Clúster RPi. Fuente: Elaboración propia.	56
Ilustración 14: Disposición de RPi's. Fuente: Elaboración propia.	56
Ilustración 15: Conexión de Red entre RPi's. Fuente: Elaboración propia.	57
Ilustración 16: Alimentación Cluster. Fuente: Elaboración propia.	58
Ilustración 17: Alimentación RPi's. Fuente: Elaboración propia.	58
Ilustración 18: Cables de Alimentación. Fuente: Elaboración propia.....	59
Ilustración 19: Datacenter DMH - Servidor de Pruebas. Fuente: Elaboración propia.....	59

Ilustración 20: Configuración del dominio centrado en Encarnación.	
Fuente: Elaboración propia.....	69
Ilustración 21: Grillas del dominio configurado. Fuente: Elaboración propia.	70
Ilustración 22: GEMHA - Procesamiento de datos. Fuente: Elaboración propia.....	74
Ilustración 23: GEMHA - Proceso de Recolección y Procesamiento de Datos.	
Fuente: Elaboración propia.....	77
Ilustración 24: GEMHA - Ingreso al sistema. Fuente: Elaboración propia.	78
Ilustración 25: GEMHA - Tipos de estaciones. Fuente: Elaboración propia.	78
Ilustración 26: GEMHA - Gestión de Estaciones. Fuente: Elaboración propia.	78
Ilustración 27: GEMHA - Datos de Estaciones. Fuente: Elaboración propia.	79
Ilustración 28: GEMHA - Página de inicio. Fuente: Elaboración propia.....	80
Ilustración 29: GEMHA - Página Estación. Fuente: Elaboración propia.....	81
Ilustración 30: WRF Runner – Automatización de Simulaciones.	
Fuente: Elaboración propia.....	83
Ilustración 31: WRF Runner - Dashboard. Fuente: Elaboración propia.	84
Ilustración 32: WRF Runner - Simulaciones. Fuente: Elaboración propia.	85
Ilustración 33: WRF Runner - Detalles de Simulación.	
Fuente: Elaboración propia.....	85
Ilustración 34: WRF Runner - Sitios. Fuente: Elaboración propia.	86
Ilustración 35: Esquema conceptual del Servidor Web y la API.	
Fuente: Elaboración propia.....	86
Ilustración 36: APP - Pronóstico de distrito. Fuente: Elaboración propia.	87
Ilustración 37: APP - Selección de distritos. Fuente: Elaboración propia.	87
Ilustración 38: APP - Pantalla inicial. Fuente: Elaboración propia.....	88
Ilustración 39: GEMHA - Job Age y Queue Size. Fuente: Elaboración propia.....	90
Ilustración 40: Primer Clúster RPi. Fuente: Elaboración propia.	91
Ilustración 41: Compilación WRF en Arquitectura ARM.	
Fuente: Elaboración Propia.	93
Ilustración 42: Proceso matado por falta de memoria.	
Fuente: Elaboración propia.....	96
Ilustración 43: Validación del Modelo WRF - Temperatura - Día 1.	
Fuente: Elaboración propia.....	105

Ilustración 44: Validación del Modelo WRF - Temperatura - Día 2.	
Fuente: Elaboración propia.....	105
Ilustración 45: Validación del Modelo WRF - Temperatura - Día 3.	
Fuente: Elaboración propia.....	105
Ilustración 46: Validación del Modelo WRF - Presión Atmosférica - Día 1.	
Fuente: Elaboración propia.....	106
Ilustración 47: Validación del Modelo WRF - Presión Atmosférica - Día 2.	
Fuente: Elaboración propia.....	106
Ilustración 48: Validación del Modelo WRF - Presión Atmosférica - Día 3.	
Fuente: Elaboración propia.....	106
Ilustración 49: Validación del Modelo WRF - Dirección del Viento - Día 1.	
Fuente: Elaboración propia.....	107
Ilustración 50: Validación del Modelo WRF - Dirección del Viento - Día 2.	
Fuente: Elaboración propia.....	107
Ilustración 51: Validación del Modelo WRF - Dirección del Viento - Día 3.	
Fuente: Elaboración propia.....	107
Ilustración 52: Validación del Modelo WRF - Velocidad del viento - Día 1.	
Fuente: Elaboración propia.....	108
Ilustración 53: Validación del Modelo WRF - Velocidad del viento - Día 2.	
Fuente: Elaboración propia.....	108
Ilustración 54: Validación del Modelo WRF - Velocidad del viento - Día 3.	
Fuente: Elaboración propia.....	108
Ilustración 55: Validación del Modelo WRF - Precipitación - Día 1.	
Fuente: Elaboración propia.....	109
Ilustración 56: Validación del Modelo WRF - Precipitación - Día 2.	
Fuente: Elaboración propia.....	109
Ilustración 57: Validación del Modelo WRF - Precipitación - Día 3.	
Fuente: Elaboración propia.....	109
Ilustración 58: Clúster RPi's - Mediciones. Fuente: Elaboración propia.	115
Ilustración 59: Clúster RPi's - Corte láser MDF. Fuente: Elaboración propia.	116
Ilustración 60: Clúster RPi's - Verificación de medidas y cableado.	
Fuente: Elaboración propia.....	116
Ilustración 61: Clúster RPi's - Pintura. Fuente: Elaboración propia.....	117

Ilustración 62: Clúster RPi's - Cableado de alimentación.	
Fuente: Elaboración propia.....	117
Ilustración 63: Clúster RPi's -Cableado de Red. Fuente: Elaboración propia.....	118
Ilustración 64: Clúster RPi's - Funcionamiento. Fuente: Elaboración propia.	118
Ilustración 65: Nota sobre carencia de un modelo de predicción numérica del tiempo para Encarnación. Fuente: DMH – DINAC.	119
Ilustración 66: Nota validación del modelo WRF. Fuente: DMH – DINAC.	120
Ilustración 67: Disertación MSc. Lic. Julián Báez. Fuente: Elaboración propia.	121
Ilustración 68: Disertación Ever Barreto y Marico Tamada.	
Fuente: Elaboración propia.....	121
Ilustración 69: Taller Hidrometeorología Aplicada - Exposición Víctor Álvarez DMH.	
Fuente:Elaboración propia.....	122
Ilustración 70: Taller Hidrometeorología Aplicada – Asistentes.	
Fuente:Elaboración propia.....	122
Ilustración 71: Estaciones convencional y automática - Aeropuerto de Encarnación.	
Fuente: Elaboración propia.....	123
Ilustración 72: Observador de la DINAC presentando EMA – Aeropuerto de Encarnación. Fuente: Elaboración propia.....	123
Ilustración 73: Visita técnica a la DMH - Estación Automática OTT.	
Fuente: Elaboración propia.....	124
Ilustración 74: Visita técnica a la DMH - Datacenter. Fuente: Elaboración propia...	124
Ilustración 75: Servidor de prueba DMH. Fuente: Elaboración propia.....	125
Ilustración 76: Servidor de prueba en nuevo Rack. Fuente: Elaboración propia.	125
Ilustración 77: Nicolás Rodríguez instalando UPS y PDU.	
Fuente: Elaboración propia.....	126
Ilustración 78: Ing. Wilson Caballero transportando el nuevo Rack.	
Fuente: Elaboración propia.....	126
Ilustración 79: Nuevo Rack de la DMH instalado. Fuente: Elaboración propia.	126

Índice de Tablas

Tabla 1: Velocidades de Tarjetas SD	51
Tabla 2: Requerimientos de energía del Raspberry Pi	54
Tabla 3: Componentes principales del clúster.....	55
Tabla 4: Parametrizaciones del Dominio	71
Tabla 5: Tiempos de ejecución del Modelo WRF en el Clúster de RPi's para un periodo de 6 horas.....	96
Tabla 6: Parametrización por defecto del WRF 3.8.1	97
Tabla 7: Parametrización del WRF - Prueba 2.....	98
Tabla 8: Parametrización del WRF - Prueba 3.....	98
Tabla 9: Parametrización del WRF - Prueba 5.....	99
Tabla 10: Parámetros físicos WRF.....	100
Tabla 11: Tiempo de ejecución del Modelo WRF en el Clúster de RPi's para un periodo de 72 horas.....	100
Tabla 12: Tiempos de ejecución del Modelo WRF en el Servidor de Pruebas.....	103
Tabla 13: Listado de estaciones automáticas. Red DMH.....	113
Tabla 14: Datos en bruto EHAS - DMH	113
Tabla 15: Datos en bruto EMAS - DMH.....	115
Tabla 16: Costos del proyecto	115

Resumen

El presente trabajo de investigación tiene como propósito principal, la previsión meteorológica para la ciudad de Encarnación utilizando el Modelo de Investigación y Pronóstico del Clima (WRF) en un clúster de Raspberry Pi's.

El proyecto surge como una respuesta a la necesidad de contar con previsiones meteorológicas enfocadas en la ciudad de Encarnación, buscando proveer de esta forma a la sociedad civil, de una herramienta que permita anticipar fenómenos atmosféricos que tienen una incidencia directa en la vida de las personas y en la protección de la propiedad privada.

Este proyecto incluye varios aspectos principales como la construcción de un Clúster de Raspberry Pi's, el ajuste del Modelo WRF para su ejecución en estos computadores, la automatización de la ejecución del modelo, el pos procesamiento de los datos y finalmente, la visualización de los pronósticos generados en la forma de una aplicación móvil.

Palabras clave: WRF, Clúster, Raspberry Pi, ARM

Abstract

The primary goal of this research paper is the weather forecasting for the city of Encarnación, using the Weather Research and Forecasting Model (WRF), on a Cluster of Raspberry Pi.

The project is a response to the need to have weather forecasts focused on the city of Encarnación, seeking to provide the society with a tool to anticipate atmospheric phenomena that have a direct impact on life and private property.

This project includes several key aspects such as the build of a cluster of Raspberry Pi's, the adjustment of the WRF Model for execution on these computers, the automation of the model execution, the post processing of the data and, finally, the visualization of the generated forecasts through a mobile application.

Keywords: WRF, Cluster, Raspberry Pi, ARM

1 Introducción

1.1 Definición del tema

Automatización de la recolección y generación de pronósticos del tiempo para la ciudad de Encarnación.

1.2 Planteamiento general

A nivel mundial la predicción del tiempo ha evolucionado notablemente, se utilizan herramientas que registran el estado atmosférico de momento a momento. La información obtenida es estandarizada y almacenada en centros meteorológicos mundiales.

Los países desarrollados cuentan con satélites y una vasta red de sensores y radares que cubren la totalidad de su territorio; obteniendo de forma automática los datos necesarios para la predicción del tiempo. Así también, con el poder computacional capaz de procesar estos datos.

El Paraguay cuenta con una red de estaciones meteorológicas en constante crecimiento, pero no se utiliza ningún sistema completamente automatizado de recolección de datos y de predicción del tiempo. Se realiza a través de la observación de imágenes satelitales, pronósticos globales y/o de países vecinos.

Puntualmente en la ciudad de Encarnación y ciudades vecinas no se cuenta con ningún proceso de generación automática de pronósticos del tiempo operativos para la región (ver punto 12.4).

Se plantea realizar la automatización de la recolección de datos y la generación de pronósticos del tiempo para la ciudad de Encarnación.

Este trabajo de investigación pretende automatizar por una parte la obtención y el pre procesamiento de los datos requeridos, generar los pronósticos especializados para Encarnación y por último, pos procesar los datos generados para poder ser utilizados en una aplicación móvil.

1.3 Problemas a resolver

Actualmente la ciudad de Encarnación no cuenta con ningún sistema operativo de predicción meteorológica del tiempo enfocado en la zona (ver punto 12.4).

Los pronósticos generados por el Servicio Meteorológico Nacional para el interior se realizan abarcando una amplia zona (departamental), no pudiendo representar fenómenos puntuales que se dan en la ciudad.

La carencia de un método automatizado de predicción del tiempo a nivel regional, ocasiona retardo y falta de precisión en el pronóstico del tiempo, lo cual obliga a recurrir a datos otorgados por instituciones extranjeras, las que no están especializadas en la región (ej. Weather Channel, Yahoo Weather, Wunderground, etc.).

Por otra parte, la red de estaciones meteorológicas a cargo de la Dirección de Meteorología e Hidrología (DMH), no cuenta con una base de datos unificada ni con una interface de acceso programático a estos registros (API).

Para generar los pronósticos del tiempo se deben procesar una gran cantidad de datos, por lo que es necesario contar con un gran poder de procesamiento.

1.4 Fundamentación

Gracias a la centralización de los datos obtenidos de las estaciones meteorológicas automáticas de todo el país, se podrá contar con acceso programático a estos datos.

La ejecución de un modelo de predicción del tiempo para la ciudad de Encarnación a una mayor resolución puede brindar pronósticos más precisos.

Los dispositivos móviles son de uso común hoy en día y permite a la población a tener acceso inmediato a datos del tiempo.

1.5 Objetivo general

Implementar el Modelo de Investigación y Pronóstico del Tiempo en un clúster de Raspberry Pi's para la previsión meteorológica del tiempo en la zona de Encarnación.

1.6 Objetivos específicos

- Implementar el clúster de Raspberry Pi's.
- Determinar y recopilar datos para alimentar al modelo.

- Analizar datos y ajustarlos al modelo de entrada de datos del WRF.
- Determinar los parámetros de adaptabilidad del modelo WRF a la zona de estudio.
- Adaptar el modelo WRF para su ejecución en el clúster.
- Analizar y comparar los datos devueltos por el modelo WRF.
- Contrastar datos obtenidos con observaciones de campo.
- Determinar necesidades de iteración en base a los datos comparados.
- Graficar resultados utilizando herramientas Open Source existentes.
- Validar resultados utilizando estándares establecidos.
- Poner en funcionamiento un servidor de datos generados por el modelo.
- Desarrollar una aplicación móvil para visualizar los resultados.

1.7 Alcance

Para este proyecto de investigación se pretende realizar el pronóstico del tiempo por medio del modelo WRF-ARW para la zona del centro de Encarnación ejecutándolo en un clúster de Raspberry Pi's.

1.8 Contenido del libro

La tesis se encuentra organizada en 14 capítulos, cuyo contenido se resume a continuación:

El capítulo 1 trata sobre la definición del tema, la fundamentación, objetivos de la tesis y su alcance.

El capítulo 2 contiene una introducción a la Meteorología, sus orígenes y conceptos.

El capítulo 3 hace una introducción a la predicción numérica del tiempo, su historia y explica sus procesos y conceptos.

El capítulo 4 habla sobre el Modelo de Investigación y Pronóstico del Tiempo (WRF) y hace una breve reseña. Explica además, sus características principales y motores disponibles.

El capítulo 5 hace una introducción a los clústeres de computadoras y explica sus diferentes componentes.

El capítulo 6 detalla el hardware utilizado, mientras que el Capítulo 7 habla del Software empleado en la presente investigación.

El capítulo 8 describe las configuraciones, parametrizaciones y ajustes realizados al WRF.

El capítulo 9 explica en detalle los distintos sistemas desarrollados en el contexto de este trabajo de investigación.

El capítulo 10 detalla las distintas pruebas realizadas en cada etapa de la investigación.

El capítulo 11 realiza una conclusión del trabajo desarrollado y define posibles líneas de investigación a futuro.

Los capítulos 12, 13 y 14 corresponden al anexo, glosario de términos y bibliografía respectivamente.

2 Meteorología

2.1 Orígenes

El interés por el tiempo atmosférico se remonta a los comienzos de la humanidad y los intentos por comprenderlo y predecirlo a los primeros tiempos de los que tenemos testimonios escritos. Desde el principio el conocimiento de los fenómenos atmosféricos estuvo mediatizado por creencias sobrenaturales y asociado al movimiento de los astros. Tardaría mucho tiempo, casi hasta nuestros días, en emanciparse de estas dos influencias. Los primeros intentos de explicarlos bajo una aproximación científica corresponden a la civilización griega y la obra de Tales de Mileto (624 – 545 a. C.), Anaximandro (611 – 547 a. C.), Hipócrates de Cos (460 – 375 a. C.), mucho más conocido como médico, y otros sabios helénicos. La Torre de los Vientos, bajo la colina de la Acrópolis de Atenas, es un testimonio que ha llegado a nuestros días del conocimiento griego sobre la relación entre el viento y otros elementos del tiempo.

Pero sin duda, el estudio más amplio y difundido de aquel entonces corresponde a Aristóteles (384 – 322 a. C.). Debemos al filósofo macedonio la introducción del término “Meteorología”, construido a partir de las palabras griegas Meteoros “alto en el cielo” y lógica, “conocimiento, tratado”. Además a él se debe el primer tratado dedicado principalmente a la atmósfera, los cuatro libros conocidos como “los Meteorológicos” que Aristóteles escribió probablemente hacia el 340 a. C. En realidad Aristóteles entendía el término en un sentido más amplio: *“todos los efectos que se pueden llamar comunes al aire y agua y las formas y partes de la tierra y los efectos de sus partes”*, un programa que no desentona con el enfoque actual de la ciencia atmosférica [1].

2.2 La época instrumental y el desarrollo de la navegación

Los pronósticos del tiempo basados en relaciones astrológicas alcanzaron gran popularidad al final de la Edad Media y mantuvieron una extendida aceptación hasta muy avanzado el siglo XIX. La secular tradición de reglas sobre el tiempo

introducidas por los agricultores y refranes populares, completaban el breve panorama de conocimientos meteorológicos al inicio del siglo XVIII.

Los primeros instrumentos para medir las variables meteorológicas como el termómetro (Galileo, 1607), el barómetro (Torricelli, 1643) y el anemómetro (Hooke, 1667) así como el auge de la navegación permitieron incluir la experimentación objetiva en el estudio de la atmósfera. Durante los siglos XVII y XVIII se empezó a desvelar la relación de consecuencia entre las condiciones del tiempo en lugares diferentes de la Tierra. Las observaciones de marinos como Edmond Halley o James Cook y la intuición de hombres de ciencia como George Hadley, Antoine Lavoiser o Benjamin Franklin fueron ofreciendo explicaciones cada vez más consistentes sobre la circulación general de la atmósfera.

La consecuencia natural de esa noción espacial de la evolución atmosférica fue la ambición de crear redes de observatorios que ofreciesen datos a lo largo de áreas extensas. En el siglo XVIII, con la invención del telégrafo eléctrico, se desarrolla la primera red nacional de observaciones en Inglaterra.

A principio del siglo XIX se comprendió la utilidad de representar los campos de presión y temperatura en los mapas y emplear cierta simbología para otros datos observados como el viento o la precipitación. Nació así la meteorología sinóptica que se ocupa de la representación en mapas del estado de la atmósfera, en principio y hasta mucho después, solamente sobre la superficie terrestre [1].

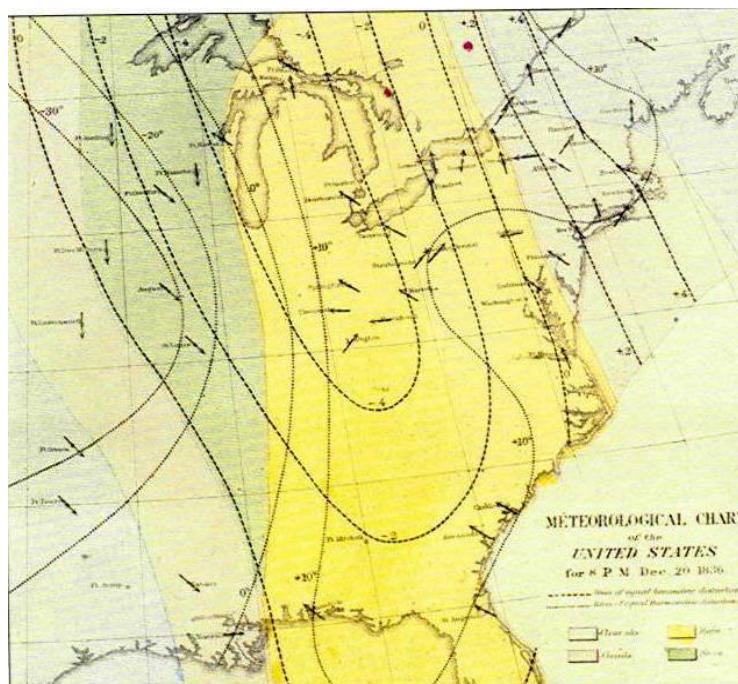


Ilustración 1: Mapa de depresión sobre el este de Estados Unidos en 1842.
Fuente: Elias Loomis.

2.3 La institucionalización de la meteorología en el siglo XIX

Las observaciones meteorológicas simultáneas y su representación sinóptica servían para la investigación científica, pero no para informar sobre la evolución atmosférica a tiempo de que fuera útil. Pero el desarrollo del telégrafo por Samuel Morse hacia 1840 fue enseguida aprovechado por los meteorólogos para transmitir y recopilar con rapidez las observaciones. Así aparecieron a partir de 1850, los primeros servicios meteorológicos organizados como instituciones capaces de producir y difundir información y predicción del tiempo. Alcanzaron un rápido éxito en los países desarrollados y recibieron pronto el apoyo de los gobiernos por su utilidad para la navegación marítima y progresivamente para otros usuarios.

Hubo enseguida necesidad de coordinar internacionalmente la realización de observaciones y su transmisión por las líneas telegráficas y aunque supuso cierto tiempo y dificultades, la colaboración internacional estaba razonablemente organizada en el último cuarto del siglo XIX. En septiembre de 1873 se reunió en Viena el primer Congreso Meteorológico Internacional, que dio origen al primer organismo de coordinación a escala mundial, la Organización Meteorológica Internacional. Desde entonces la meteorología ha sido uno de los campos de cooperación internacional más comprometida y organizada.

A pesar de ese éxito de partida la predicción meteorológica no tardó en mostrar grandes limitaciones; la atmósfera desafiaba la mayoría de los pronósticos sobre su evolución a partir de las 24 horas de su producción y a menudo antes. En realidad los pobres resultados que ofrecía la predicción meteorológica respondían a la propia limitación del conocimiento sobre la atmósfera y la física del aire que contrastaban con el avance alcanzado en otras ramas de la física a final del siglo XIX. La meteorología se había desarrollado hasta entonces como una ciencia principalmente empírica basada en la observación de los fenómenos y básicamente de la observación de las variables observadas en la superficie de la tierra. Por supuesto, se suponía que la evolución atmosférica dependía de las leyes físicas, pero no se había profundizado demasiado en su interpretación. Los meteorólogos confiaban principalmente en la observación y la experiencia [1].

2.4 La meteorología moderna

Al comienzo del siglo XX un científico excepcional, el noruego Vilhelm Bjerknes (1862-1951), se dedicó a adecuar la meteorología práctica como una aplicación de la física atmosférica. Entre otras importantes contribuciones, en 1904 propuso enfocar la

predicción del tiempo como un problema de valores iniciales partiendo de la observación tridimensional de la atmósfera, lo que constituye el fundamento de la predicción actual.

Bjerknes expuso con clarividencia las dos principales dificultades para llevar a cabo tal programa: la dificultad del problema matemático y la escasez e inadecuación de las observaciones meteorológicas de la época; los datos de que se disponía no eran suficientes y además se restringían a la superficie de la Tierra. A principios del siglo XX apenas había comenzado a desarrollarse la observación de la atmósfera superior mediante globos sonda y cometas que portaban un meteorógrafo para registrar observaciones. El seguimiento de los globos con teodolitos desde tierra permitía deducir la dirección y velocidad del viento. El desarrollo de la aviación que también precisaba de esos datos facilitó en las décadas siguientes profundizar en el conocimiento de los niveles altos de la atmósfera.

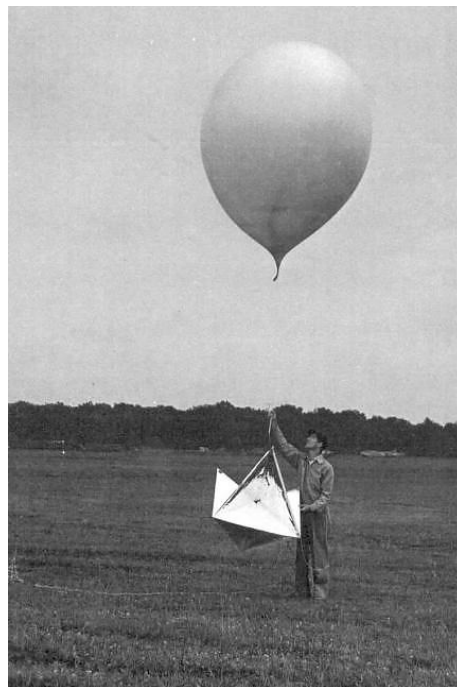


Ilustración 2: Globo sonda. Fuente: Servicio Meteorológico de los Estados Unidos

En los años veinte Bjerknes hizo otro aporte fundamental junto con un equipo de jóvenes colaboradores, su hijo Jakob Bjerknes, Carl Gustav Rossby, Thor Bergeron, Havor Solberg y otros, desarrollaron con detalle la teoría de la evolución de las depresiones de latitudes medias y sus masas de aire y frentes asociados.

Los grandes avances en la comprensión de la evolución atmosférica y el desarrollo de las técnicas de observación en esos años no fueron sin embargo correspondidos con un aumento sensible del horizonte temporal de la predicción del tiempo. Pero en 1920 un matemático meteorólogo inglés, Lewis Fry Richardson, se enfrentó a la otra dificultad que Bjerkes había indicado en su programa de 1904: El problema matemático de resolver las ecuaciones de la evolución atmosférica que abordó mediante los llamados métodos numéricos. Aún así no era posible en aquella época realizar en tiempo útil los inmensos cálculos necesarios y, al darse cuenta de la imposibilidad práctica, Richardson escribió que “quizá algún día los avances de la computación la hagan más rápida que el tiempo atmosférico y con un coste menor que los gastos implicados” sugiriendo que tal cosa acaecería antes de unos 25 años. Tal como predijo Richardson, los métodos “numéricos” que él sugirió fueron esencialmente los empleados cuando, a partir de las experiencias de Charney y Von Neuman en 1950, los primeros

ordenadores empezaron a realizar rápidamente los cuantiosísimos cálculos necesarios y dieron su nombre a las técnicas de predicción “numérica” utilizada de forma general en nuestros días [1].

Mientras tanto, la colaboración internacional en meteorología, iniciada en el siglo XIX, no dejó de crecer. Tanto el progreso de la ciencia y las técnicas meteorológicas como el propio desarrollo de los usuarios, principalmente la aviación, obligaron a disponer de servicios meteorológicos cada vez más profesionales y a una cooperación comprometida con empleo de recursos amplios y costos cuya disposición no estaba fácilmente en manos de muchas de esas instituciones. Tras la Segunda Guerra Mundial hubo un consenso unánime para reestructurar la cooperación meteorológica internacional y se aprobó la creación de un nuevo organismo con carácter intergubernamental, la Organización Meteorológica Mundial (OMM). Su convenio entró en vigor en 1950 y desde entonces la OMM ha tenido un papel fundamental en la regulación de las actividades meteorológicas y climatológicas en todo el planeta, apoyado en el papel de los llamados Servicios Meteorológicos Nacionales de cada país para llevar a cabo sus programas (Dirección de Meteorología e Hidrología en Paraguay).

Con el lanzamiento del primer satélite artificial en 1957, no tardó en ponerse a manifiesto la utilidad de la observación de la atmósfera desde el espacio. La NASA de EEUU lanzó el 1 de abril de 1960 el satélite TIROS-1 que funcionó durante 78 días enviando imágenes de la atmósfera y considerado como el primer satélite meteorológico con éxito. Durante las décadas siguientes la observación desde el espacio se convirtió en una de las dos herramientas fundamentales para la meteorología moderna junto con los modelos físico-matemáticos de la atmósfera, ejecutados por ordenadores cada vez más potentes, que a su vez fueron aumentando drásticamente su fiabilidad gracias al uso de la observación desde el espacio como datos de partida. Después otras tecnologías de teledetección como los radares han hecho aumentar exponencialmente el conjunto de datos disponibles para alimentar los modelos y vigilar los fenómenos atmosféricos.

Actualmente, la predicción operativa del tiempo se basa en modelos numéricos de simulación de la atmósfera, ejecutados por potentes ordenadores a partir de la observación de los datos atmosféricos provenientes de muchas fuentes, sobre todo de satélites, que se introducen en los modelos con una serie de técnicas muy desarrolladas.

2.5 Tiempo y Clima

El tiempo atmosférico o meteorológico (id. inglés Weather) es el conjunto de parámetros que en un momento dado y en un lugar determinado caracterizan el estado atmosférico. Es un estado pasajero.

Por otra parte, se denomina Clima (id. inglés Climate) al conjunto de fenómenos de tipo meteorológico que tienen lugar en una determinada región a lo largo del tiempo. Es un valor permanente, es así que se puede hablar de un tiempo promedio para un determinado lugar. Se define por lo general para un mes o una estación del año, y considera el promedio de los datos del tiempo de 30 años (en algunas circunstancias el plazo puede ser de 10 años).

De esta forma, la Meteorología estudia el tiempo y su predicción a corto plazo (de uno a varios días), mientras que la Climatología analiza el clima y su predicción a largo plazo con carácter permanente.

2.6 La Atmósfera terrestre

La atmósfera es la capa de gas (principalmente nitrógeno y oxígeno) que rodea la tierra. [2]

En comparación con el diámetro aproximado de la tierra (12.000 kilómetros), la capa atmosférica es muy delgada, cerca del 99 por ciento de todo el gas atmosférico está concentrado en los primeros 30 kilómetros desde la superficie terrestre. [2]



Ilustración 3: Atmósfera terrestre.
Fuente: Taller de Hidrometeorología UCI - 2016

Sin embargo, esta fina capa de gas protege la vida de la Tierra, absorbiendo en la capa de ozono parte de la radiación ultravioleta proveniente del sol y escombros espaciales (meteoritos). Además, almacena energía térmica gracias a la cual la Tierra es habitable, tanto para el ser humano como para las demás formas de vida.

2.7 Capas de la Atmósfera

La atmósfera se estructura en capas definidas por los cambios de temperatura que se producen con la altitud. En orden ascendente estas capas son:

- **Tropósfera:** es la capa más baja, en la que se desarrolla la vida y la mayoría de los Fenómenos meteorológicos. Se extiende hasta una altura aproximada de 10 km en los polos y 18 km en el ecuador. En la troposfera la temperatura disminuye paulatinamente con la altura hasta alcanzar los -70°C . Su límite superior es la tropopausa.
- **Estratósfera:** en esta capa la temperatura se incrementa hasta alcanzar aproximadamente los -10°C a unos 50 km de altitud. Es en esta capa donde se localiza la máxima concentración de ozono, “capa de ozono”, gas que al absorber parte de la radiación ultravioleta e infrarroja del Sol posibilita la existencia de condiciones adecuadas para la vida en la superficie de la Tierra. El tope de esta capa se denomina estratopausa.
- **Mesósfera:** en ella, la temperatura vuelve a disminuir con la altura hasta los -140°C . Llega a una altitud de 80 km, al final de los cuales se encuentra la mesopausa.
- **Termósfera:** es la última capa, que se extiende hasta varios cientos de kilómetros de altitud, presentando temperaturas crecientes hasta los 1000°C . Aquí los gases presentan una densidad muy baja y se encuentran ionizados.

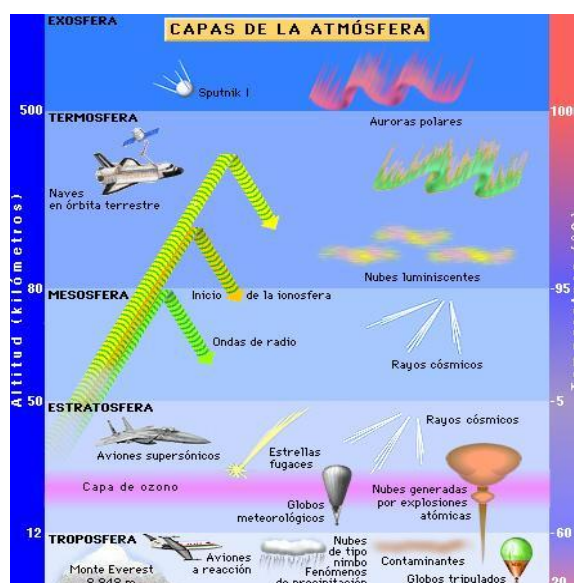


Ilustración 4: Variación térmica en la atmósfera.
Fuente: Taller de Hidrometeorología UCI - 2016



Ilustración 5: Capas de la atmósfera. Fuente: Taller de Hidrometeorología UCI - 2016

2.8 Vapor de Agua

La atmósfera contiene además vapor de agua, un componente de enorme importancia que contribuye a la formación de las nubes de agua y de hielo que generan los distintos tipos de precipitación. El vapor de agua cambia de estados continuamente: lluvia, granizo y allí se absorbe y libera energía también almacena y libera grandes cantidades de energía térmica denominada calor latente que constituye la fuente de energía que estimula el desarrollo de tormentas y huracanes.

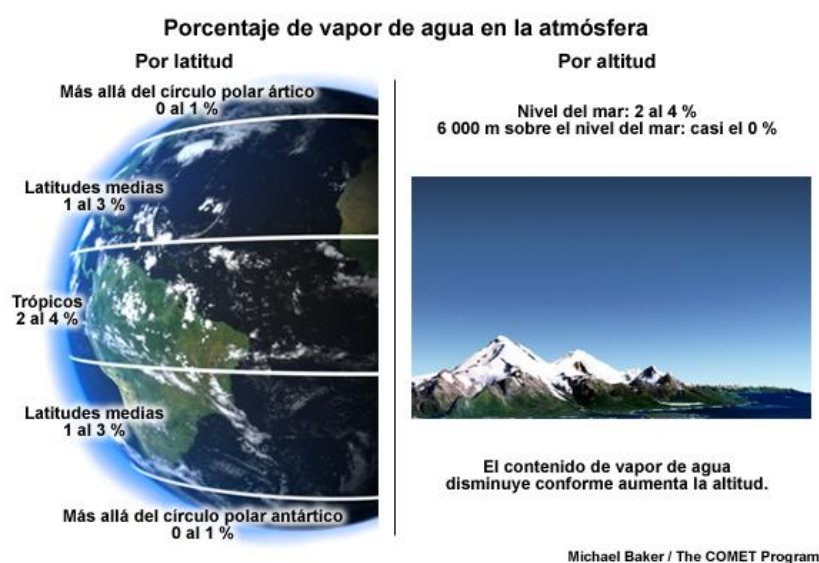


Ilustración 6: Vapor de agua en la atmósfera.
Fuente: Taller de Hidrometeorología UCI - 2016

2.9 Meteoros

Un meteoro es un fenómeno observado en la atmósfera o sobre la superficie terrestre consistente en: una suspensión, una precipitación, un depósito o un fenómeno natural óptico o eléctrico.

Los meteoros se pueden clasificar en: Hidrometeoros, Litometeoros, Fotometeoros y Electrometeoros:

Hidrometeoros: son meteoros formados por un conjunto de partículas acuosas, líquidas o sólidas que caen a través de la atmósfera, están en suspensión en ella, son remontadas de la superficie de la Tierra por el viento o son depositadas sobre objetos situados en la superficie o en la atmósfera libre. *Ej. Llovizna, lluvia, chubasco, nieve, granizo, nubes, niebla, neblina, etc.*

Litometeoros: Meteoros consistentes en un conjunto de partículas, la mayoría de las cuales son sólidas y no acuosas. Las partículas están más o menos suspendidas en el

aire o son levantadas desde el suelo por el viento. Ej. *Calima, tempestad de polvo o arena, humo, tolvaneras, etc.*

Fotometeoros: Son fenómenos luminosos que se producen en la atmosfera terrestre por diversas propiedades de la radiación luminosa. Es decir, ocurren cuando se producen efectos especiales en la atmósfera con la luz. Ej. *Corona solar, halo solar, parhelio, arcoíris, etc.*

Electrometeoros: Manifestaciones visibles o audibles de la electricidad atmosférica. Ej. Relámpagos, rayos, fuego de San Telmo, auroras polares.

2.10 Escalas Meteorológicas

En meteorología se manejan cuatro escalas principales dependiendo del tamaño de los sistemas estudiados

- **Escala Planetaria:** trata los sistemas de alcance global. Ej. El Niño oscilación del sur.
- **Escala Sinóptica:** cubre porciones horizontales en el orden de 1.000 a 2.500 kilómetros de amplitud. Ej. Ciclones extra tropicales.
- **Mesoescala:** trata sistemas que oscilan de 6 a varios centenares de kilómetros, como ser los frentes fríos, cálidos y ocluidos, la brisa del mar, etc.
- **Microescala:** estudia los fenómenos atmosféricos de corto tiempo y menor envergadura que los de mesoescala, que generalmente cubren porciones menores a los 2 km. de amplitud. Ej. Tornado, tromba marina, pequeñas tormentas de verano.

3 Predicción Numérica del Tiempo

3.1 Breve introducción

La predicción numérica del tiempo utiliza modelos matemáticos de la atmósfera y de los océanos para pronosticar el tiempo basado en las condiciones atmosféricas actuales.

Los avances registrados en la simulación por computadora, en la década de 1950, permitió que la predicción numérica genere resultados realistas. Un sinnúmero de modelos de pronóstico tanto globales como regionales son ejecutados en diferentes países en todo el mundo, utilizando observaciones de la atmósfera basados en radiosondas, satélites meteorológicos y otros sistemas de observación como entrada de datos.

Modelos matemáticos basados en los mismos principios físicos pueden ser utilizados para generar tanto pronósticos de tiempo a corto plazo como pronósticos climáticos de largo plazo.

Procesar los grandes conjuntos de datos y realizar los complejos cálculos necesarios para la predicción numérica del tiempo requiere potentes supercomputadoras. Incluso con el creciente poder de cómputo de las supercomputadoras, la precisión de la predicción numérica del tiempo se extiende solamente a unos seis días. Factores que afectan la precisión de la predicción incluye la densidad y calidad de las observaciones utilizadas como entrada al modelo, junto con las deficiencias de los mismos modelos.

Un problema más fundamental radica en la naturaleza caótica de las ecuaciones diferenciales parciales que gobiernan la atmósfera. Es imposible resolver las ecuaciones correctamente, y errores pequeños crecen en el tiempo (duplicándose aproximadamente cada cinco días). El conocimiento actual indica que este comportamiento caótico limita la precisión de la predicción a aproximadamente 14 días, incluso con datos de entrada precisos y un modelo sin defectos. Adicionalmente las ecuaciones diferenciales parciales utilizadas en el modelo deben de ser complementadas con parametrizaciones de radiación solar, procesos húmedos (nubes y

precipitación), intercambio de calor, suelo, vegetación, agua superficial y los efectos del terreno.

3.2 Historia

La historia de la predicción empezó en 1920 con Lewis Fry Richardson, quien utilizó los procedimientos desarrollados por Vilhelm Bjerknes para producir a mano una predicción de seis horas del estado de la atmósfera sobre dos puntos en Europa central, el cual llevó seis semanas realizarlo [3] [4]. No fue hasta la llegada de la computadora y la simulación por computadora que el tiempo de cómputo se redujo a menos que el periodo de predicción en sí.

El ENIAC fue utilizado para crear las primeras predicciones por computadora en 1950, basada en una muy simplificada aproximación de las ecuaciones que gobiernan la atmósfera [5]. En 1954, el grupo de Carl-Gustav Rossby en el Instituto Meteorológico e Hidrológico Sueco utilizó el mismo modelo para producir la primera predicción

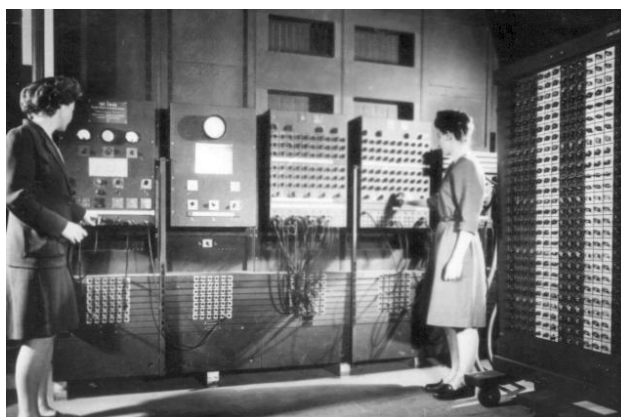


Ilustración 7: Eniac. Fuente: U. S. Army Photo.

operacional (predicción de rutina para usos prácticos) [6]. La predicción numérica operacional del tiempo en los Estados Unidos se inició en 1955, bajo la Joint Numerical Weather Prediction Unit (JNWPU), un proyecto en conjunto de la Fuerza Aérea, la Armada y la Oficina Meteorológica de los Estados Unidos. En 1956, Norman Philips desarrolló un modelo matemático que podía describir realísticamente patrones mensuales y estaciones de la tropósfera; Este se convirtió en el primer modelo climático [7].

A medida que las computadoras fueron creciendo en poder de cómputo, el tamaño de los conjuntos de datos iniciales se incrementó y nuevos modelos atmosféricos se desarrollaron para tomar ventaja de la nueva capacidad en cómputo disponible. Estos nuevos modelos incluyen más procesos físicos en las simplificaciones de las ecuaciones de movimiento en simulaciones numéricas de la atmósfera. En 1966, Alemania Oeste y los Estados Unidos empezaron a generar pronósticos operacionales basados en modelos de ecuaciones primitivas, seguidos por el Reino Unido en 1972 y Australia en 1977 [3] [8]. Al principio de los 80, los modelos empezaron a incluir las interacciones del suelo y la vegetación con la atmósfera, que llevó a pronósticos más realistas.

3.3 Proceso de inicialización

La atmósfera es un fluido. Como tal, la idea de la predicción numérica del tiempo es tomar una muestra del estado del fluido en un tiempo dado y, utilizando las ecuaciones de dinámica de fluidos y termodinámica, estimar el estado del fluido en un tiempo en el futuro. El proceso de ingresar los datos observados al modelo para generar las condiciones iniciales se llama "inicialización". En tierra, mapas de terreno disponibles a resoluciones a un kilómetro globalmente son utilizadas para ayudar a modelar las circulaciones atmosféricas en regiones con topografía accidentada, con el fin de representar características como descendentes de ladera, ondas de montaña y nubosidad que afecta la radiación solar entrante [9]. Las principales entradas son observaciones de dispositivos llamados radiosondas en globos atmosféricos que miden los distintos parámetros atmosféricos y lo transmiten a un receptor fijo en tierra, y también datos de satélites meteorológicos. La Organización Meteorológica Mundial reglamenta la estandarización de la instrumentación, prácticas de observación y el tiempo de estas observaciones a nivel mundial. Las estaciones (convencionales) reportan datos cada hora en formato METAR y cada seis horas en formato SYNOP. Estas observaciones están espaciadas irregularmente, por lo que son procesadas utilizando métodos de asimilación de datos y análisis objetivo, que realizan control de calidad y obtiene valores de ubicaciones usables por los algoritmos del modelo matemático [10]. Estos datos son entonces utilizados en el modelo como punto de inicio para un pronóstico.

Una gran variedad de métodos son utilizados para capturar datos de las observaciones para su uso en modelos numéricos. Los servicios meteorológicos lanzan radiosondas en globos atmosféricos que se elevan a través de la tropósfera y llegan incluso hasta la estratósfera. Datos de satélites meteorológicos son utilizados donde fuentes de datos tradicionales no están disponibles. El comercio provee reportes de pilotos sobre rutas de aviación y barcos reportan datos sobre sus rutas de navegación [11] [12].

3.4 Modelos

Un modelo atmosférico es un programa de computadora que produce información meteorológica para tiempos futuros en determinadas ubicaciones y altitudes. Dentro de cada modelo moderno se encuentra una serie de ecuaciones, conocidas como ecuaciones primitivas, que se utilizan para predecir el estado futuro de la atmósfera. Estas ecuaciones son utilizadas para evolucionar la densidad, presión, temperatura y velocidad del aire (viento) a través del tiempo. Las ecuaciones utilizadas son ecuaciones no lineales diferenciales que son imposibles de resolver con exactitud a través de métodos analíticos. Por lo tanto, los métodos numéricos obtienen soluciones

aproximadas. Diferentes modelos utilizan diferentes métodos de solución: algunos modelos globales y casi todos los regionales utilizan métodos de diferencias finitas para las tres dimensiones espaciales, mientras que otros modelos globales utilizan métodos espectrales para las dimensiones horizontal y vertical y métodos de diferencias finitas en la vertical.

Estas ecuaciones se inicializan desde el análisis de los datos y se determinan las tasas de cambio. Las tasas de cambio predicen el estado de la atmósfera en un corto tiempo en el futuro; el incremento de tiempo para esta predicción es llamado un "paso de tiempo". Este estado atmosférico futuro es luego utilizado como el punto de partida para otra aplicación de las ecuaciones predictivas buscando encontrar nuevas tasas de cambio, y estas nuevas tasas de cambio, predecir la atmósfera de nuevo en otro paso de tiempo en el futuro. Este paso de tiempo es repetido sucesivamente hasta que la solución alcanza el tiempo de pronóstico deseado. El largo del paso de tiempo elegido en el modelo está relacionado a la distancia entre puntos en la grilla de cómputo y es elegido para mantener una estabilidad numérica. El paso de tiempo para modelos globales están en el orden de decenas de minutos [13], mientras que para los modelos regionales son de entre uno y cuatro minutos [14]. Los modelos globales son ejecutados a distintos tiempos en el futuro. El Modelo Unificado de UKMET (Servicio Meteorológico del Reino Unido) es ejecutado para seis días en el futuro [15], mientras que el Sistema Integrado de Pronósticos del Centro Europeo (para pronósticos de mediano y corto plazo) y el Modelo Ambiental Multiescala del Servicio Meteorológico de Canadá [16], ambos son ejecutados para diez días en el futuro, y el Sistema Global de Pronóstico (GFS) del Centro de Modelado Ambiental (EMC USA) es ejecutado para dieciséis días en el futuro [17].

3.5 Parametrización

Algunos procesos meteorológicos ocurren a una escala muy pequeña o son muy complejos para ser incluidos explícitamente en los modelos de predicción numérica. La Parametrización es el procedimiento de representar estos procesos relacionándolos con variables en las escalas que resuelve el modelo. Por ejemplo, las grillas en modelos de tiempo y clima tienen lados que van desde los 5 kilómetros a los 300 kilómetros de largo. Una nube cúmulus típica tiene una escala de menos de 1 kilómetro y requeriría una grilla más fina que esto para representar físicamente las ecuaciones de movimiento de fluidos. Por lo tanto, los procesos que tales nubes representan son parametrizados, siguiendo distintos procesos de sofisticación. En los primeros modelos, si una columna de aire en una grilla era condicionalmente inestable (esencialmente, la parte de abajo era más cálida y húmeda que la parte de arriba) y el contenido de vapor de agua en cualquier punto dentro de la columna se saturaba

entonces se volcaría (el aire húmedo y cálido empezaría a ascender), y en el aire en la columna vertical se mezclaría. Esquemas más sofisticados reconocen que sólo algunas partes de la grilla pueden producir convección, arrastre y otros procesos pueden ocurrir. Los modelos que tienen grillas con lados de 5 a 25 kilómetros pueden explícitamente representar nubes convectivas, aunque necesitan parametrizar procesos microfísicos de nubes que ocurren a una escala más pequeña [18]. La formación de nubes de gran escala (de tipo stratus) está basada fundamentalmente en procesos físicos; se forman cuando la humedad relativa alcance un valor preestablecido.

La cantidad de radiación solar que llega al suelo, así como la formación de gotas en las nubes se producen a una escala molecular, y por lo tanto, deben ser parametrizadas antes de ser incluidas en el modelo. La resistencia atmosférica producida por las montañas también deben ser parametrizadas, ya que las limitaciones en la resolución de las curvas de nivel producen una subestimación importante de la resistencia. El ángulo del sol así como el impacto de múltiples capas de nubes son tomadas en cuenta. El tipo de suelo, tipo de vegetación y humedad del suelo, determinan cuánta radiación se traduce en calentamiento y cuánta humedad es llevada en la atmósfera adyacente; y por lo tanto, es importante parametrizar su contribución a estos procesos. [19]

3.6 Dominios

El dominio horizontal de un modelo es global, abarcando toda la tierra, o regional, abarcando sólo una parte de ella. Modelos regionales permiten el uso de una grilla más chica que los modelos globales porque el poder de cómputo está enfocado únicamente en un área específica en lugar de todo el globo. Esto permite a los modelos regionales resolver explícitamente fenómenos meteorológicos de más pequeña escala que no pueden ser representados en una grilla más grande de un modelo global. Modelos regionales utilizan un modelo global para especificar las condiciones en el borde del dominio (condiciones de borde) a modo de permitir a sistemas que quedan fuera del modelo regional a ingresar dentro del área del mismo. Incertidumbre y errores en el modelo regional pueden ser introducidos por el modelo global utilizado para las condiciones de borde, como pueden ser atribuibles al modelo regional en sí. [20]

Un dominio, a su vez, es dividido en partes rectangulares denominadas celdas. El tamaño de la celda define la resolución que el modelo utilizará para hacer la simulación. A un menor tamaño de celda se le atribuye una mayor resolución de dominio. Un dominio puede contener uno o más subdominios anidados con una mayor resolución que el dominio padre, tal como se puede observar en la Ilustración 8.

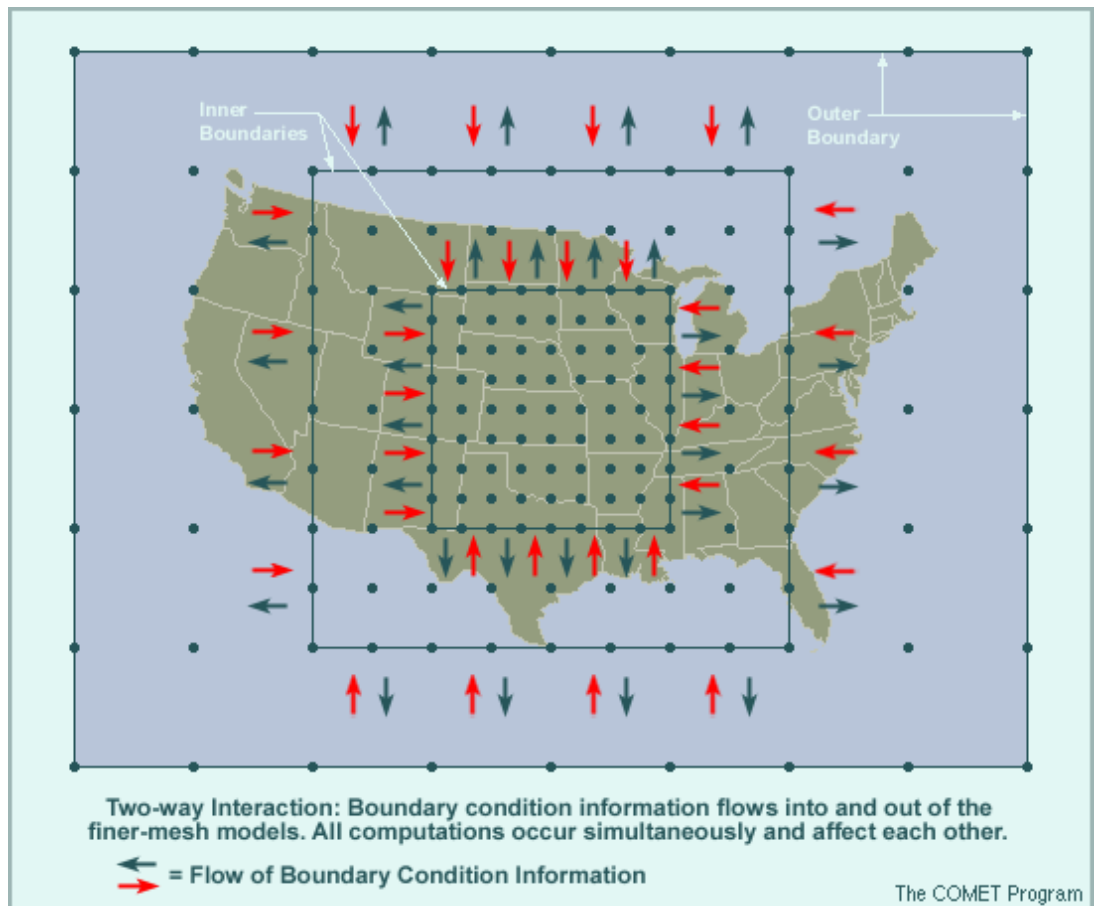


Ilustración 8: Dominio del modelo. Fuente: The Comet Program.

4 Modelo de Investigación y Pronóstico del Tiempo (WRF)

El Modelo de Investigación y Pronóstico del Tiempo es un modelo de predicción numérica del tiempo que fue diseñado para responder a necesidades de investigación y operacionales.

4.1 Reseña Histórica

El desarrollo de Este modelo tiene sus inicios al final de la década de 1990 y fue un esfuerzo colaborativo entre el Centro Nacional para la Investigación de la Atmosfera (NCAR), la Administración Nacional Oceánica y Atmosférica (NOAA) representada por el Centro Nacional para la Predicción Ambiental (NCEP) y el entonces Laboratorio de Sistemas de Pronóstico (FSL), la Agencia Meteorológica de la Fuerza Aérea (AFWA), el Laboratorio de Investigación Naval (NRL), la Universidad de Oklahoma (OU) y la Administración Federal de Aviación (FAA), todas estas, estamentos del gobierno y el ejército norteamericano.

La mayor parte del trabajo de desarrollo del modelo fue llevado a cabo principalmente por la NCAR, NOAA y AFWA.

4.2 Características principales y utilización

WRF permite a los investigadores producir simulaciones con datos reales o ideales, ofrece capacidades de predicción operativa, una plataforma flexible y robusta e incorpora avances en las partes: física, numérico-matemática y de asimilación de datos. Así también, cuenta con la contribución de numerosos desarrolladores de la comunidad científica. El modelo es actualmente utilizado operacionalmente en el Centro Nacional para la Predicción Ambiental (NCEP) y otros centros meteorológicos internacionales. Es además utilizado extensivamente para la investigación y predicción meteorológica en todo el mundo.

4.3 Motores disponibles

El modelo ofrece actualmente dos motores para el cálculo de las ecuaciones que gobiernan las atmósfera, estas variantes son conocidas como WRF-ARW (Advanced Research WRF) y WRF-NMM (Nonhydrostatic Mesoscale Model). Actualmente, el WRF-ARW es activamente desarrollado y actualizado mientras que el WRF-NMM fue descontinuado a finales del 2014. [21]

4.4 Condiciones iniciales

Como se menciona en el punto 3.6, un modelo regional debe acoplarse con uno global.

El modelo global provee las “condiciones iniciales” al modelo regional interpolando los datos del global al dominio del modelo regional y adicionalmente, sirve como “condición de borde” permitiendo que las grandes masas de aire del modelo global fluyan en los bordes laterales del dominio regional.

Los principales modelos utilizados actualmente para la predicción meteorológica a nivel global son:

- **IFS:** Sistema Integrado de Pronósticos (Integrated Forecast System) del Centro Europeo de Previsiones Meteorológicas a Mediano Plazo (European Centre for Medium-Range Weather Forecasts)
- **GFS:** Sistema Global de Pronóstico (Global Forecasting System) del Centro de Modelado Ambiental (EMC) de la Administración Nacional Oceánica y Atmosférica (NOAA)
- **UKMET:** del Servicio Meteorológico del Reino Unido (United Kingdom National Meteorological Service)
- **GEM:** (Global Environmental Multiscale Model) del Centro Meteorológico Canadiense (CMC)

De todos estos modelos, el modelo GFS de la NOAA es el único que provee salidas disponibles gratuitamente y bajo dominio público en Internet. Esto lo convierte en el modelo global más utilizado a nivel mundial ya que tanto empresas como agencias e institutos meteorológicos dependen de sus salidas para generar productos derivados.

Por esta razón, se ha decidido utilizar el GFS como condición inicial y de borde para el desarrollo de esta investigación.

4.4.1 El Modelo GFS

El GFS como ya se mencionó en el punto anterior es un modelo global gestionado por la NOAA.

Éste es ejecutado cuatro veces al día, cada seis horas, a las 00UTC, 06UTC, 12UTC y 18UTC. [22]

Las corridas de las 00UTC y 06UTC se agrupan como “Ciclo de las 00UTC” mientras que las corridas de las 12UTC y 18UTC se agrupan como “Ciclo de las 12UTC”. [23]

En la actualidad, se encuentran disponibles dos tipos de salidas:

- Una salida de alta resolución, de 192 horas (7 días)
- Una salida de baja resolución, de 192 a 382 horas (15 días)

En el modelo, la superficie terrestre se encuentra dividida en celdas de 13 km (0.25 grados, alta resolución) y 34 km (0.5 grados, baja resolución) aproximadamente. En la vertical, la atmósfera se encuentra dividida en 64 capas. [22]

A partir del 2012, la precisión del GFS empezó a decaer en comparación con el modelo IFS. El ejemplo más notable y público se dio con el Huracán Sandy que azotó la costa este de los Estados Unidos en Octubre del 2012. El IFS pronosticó correctamente que el curso del huracán ingresaría en territorio norteamericano, con varios días de anticipación, mientras que el modelo GFS pronosticó un viraje del curso hacia el océano Atlántico [24]. Es por esto que, en la actualidad, el Modelo IFS es ampliamente considerado el mejor con relación a su precisión. [25]

Afortunadamente, el Gobierno Norteamericano respondió a esta situación proveyendo más fondos para la compra de recursos de computación a la NOAA [26] y a su Servicio Meteorológico. [27]

Con esto se espera que el Modelo GFS vuelva a una posición competitiva en un corto plazo.

4.5 Componentes

El Modelo WRF consta de varios componentes, en la Ilustración 9 se pueden apreciar cada uno de ellos.

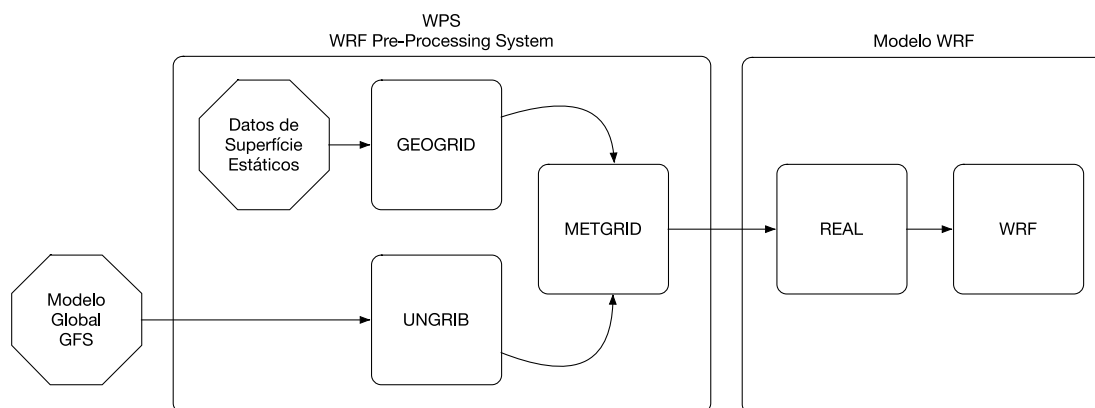


Ilustración 9: Componentes del WRF. Fuente: Elaboración propia.

Como se mencionó en el punto 4.4, el Modelo Global GFS es utilizado como condición inicial y de borde para el modelo WRF. Los archivos GFS son procesados por el sub componente UNGRIB, el cual desempaqueta los datos GRIB y los re-empaqueta en un formato intermedio que es compatible con el WPS.

Por otra parte, el GEOGRID utiliza datos de superficie que son estáticos para generar datos del terreno para la zona geográfica del dominio. Estos datos indican al modelo qué parte del terreno es, por ejemplo, tierra o agua, o si se trata de una zona boscosa, de cultivo, ciudad, etc.

Luego, el METGRID utiliza la salida del proceso GEOGRID y UNGRIB para interpolar los datos meteorológicos horizontalmente en el dominio definido.

A continuación, el REAL utiliza la salida del METGRID para interpolar los datos verticalmente a las coordenadas del modelo.

Finalmente, el WRF utiliza la salida del REAL y del METGRID para ejecutar la simulación y generar un pronóstico para el dominio configurado.

5 Clústeres de Computadoras

5.1 Breve Introducción

El término clúster proviene del idioma inglés, significa grupo, conglomerado o racimo, se aplica a los conjuntos de ordenadores unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuese una única computadora.

La tecnología de los clústeres de computadoras ha evolucionado en apoyo de actividades como ser servidores web, comercio electrónico, aplicaciones de súper cómputo, bases de datos de alto rendimiento, entre otros.

El cómputo con clústeres surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Los clústeres son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador normalmente siendo más económico que computadores individuales de rapidez y disponibilidad similares.

La construcción de los ordenadores de un clúster es bastante sencilla y económica, esto se debe a la flexibilidad que otorga, ya que pueden tener todos la misma configuración de hardware y sistema operativo (clúster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (clúster semihomogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un clúster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encarga de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

5.2 Beneficios de la tecnología clúster

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un clúster puede satisfacer estos requisitos usando los recursos que tiene asociados a él.

Los clústeres ofrecen las siguientes características a un costo relativamente bajo:

- Alto rendimiento
- Alta disponibilidad
- Alta eficiencia
- Escalabilidad

La tecnología clúster permite incrementar la capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

5.3 Clasificación de los clústeres

Los tipos de clústeres son establecidos de acuerdo con el uso que se dé y los servicios que ofrecen. Los clústeres pueden clasificarse según sus características:

Alto rendimiento (HPCC): Son clústeres en los cuales se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. Llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.

Alta disponibilidad (HACC): Son clústeres cuyo objetivo de diseño es proveer disponibilidad y confiabilidad. Estos, tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.

Alta eficiencia (HTCC): Son clústeres cuyo objetivo de diseño es ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

A pesar de las discrepancias, a nivel de requisitos de las aplicaciones; muchas de las características de las arquitecturas de hardware y software que están por debajo de las

aplicaciones en todos estos clústeres son las mismas. Inclusive, un clúster de determinado tipo, puede también presentar características de otros.

5.4 Componentes de un Clúster

En general, un clúster necesita de varios componentes de software y hardware para poder funcionar: nodos, almacenamiento, sistema operativo, conexiones de red, middleware, protocolos de comunicación y servicios, aplicaciones y ambientes de programación paralela.

5.4.1 Nodos

Pueden ser simples ordenadores, sistemas multiprocesador o estaciones de trabajo.

En informática, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. En redes de computadoras, cada una de las máquinas es un nodo.

En un clúster los nodos no disponen de teclado, ratón ni monitor y su uso está exclusivamente dedicado a realizar tareas relacionadas con el clúster.

5.4.2 Almacenamiento

El almacenamiento puede consistir en una NAS, una SAN, o almacenamiento interno en el servidor. El protocolo más comúnmente utilizado es NFS (Network File System), sistema de ficheros compartido entre servidor y los nodos. Sin embargo, existen sistemas de ficheros específicos para clústeres como Lustre (CFS) y PVFS2.

NAS (Network Attached Storage) es un dispositivo específico dedicado al almacenamiento a través de red (normalmente TCP/IP) que utiliza un sistema operativo optimizado para dar acceso a través de protocolos CIFS, NFS, FTP o TFTP.

Por su parte, DAS (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una SAN (*Storage Area Network*) a través de un canal de fibra. Estas conexiones son dedicadas.

Mientras NAS, permite compartir el almacenamiento, utilizar la red y tiene una gestión más sencilla. DAS proporciona mayor rendimiento y mayor fiabilidad al no compartir el recurso.

5.4.3 Conexiones de Red

Los nodos de un clúster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Myrinet, InfiniBand, SCI, etc.

5.4.4 Middleware

Es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un clúster lo siguiente:

- **Una interfaz única de acceso al sistema:** denominada SSI (Single System Image), genera la sensación al usuario de que utiliza un único ordenador muy potente.
- **Herramientas para la optimización y mantenimiento del sistema:** Incluye, migración de procesos, checkpoint-restart (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.
- **Escalabilidad:** debe detectar automáticamente nuevos servidores conectados al clúster para proceder a su utilización.

Existen diversos tipos de middleware, como por ejemplo: MOSIX, OpenMOSIX, Cóndor, OpenSSI, etc.

El middleware recibe los trabajos entrantes al clúster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor. Esto se realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican dónde y cómo debe distribuir los procesos, por un sistema de monitorización, el cual controla la carga de cada CPU y la cantidad de procesos en él.

El middleware también debe migrar procesos entre servidores con distintas finalidades:

- **Balanceo de la carga:** si un servidor está muy cargado de procesos, y otro está ocioso, puede transferir procesos a este último para liberar de carga al primero y optimizar el funcionamiento.
- **Mantenimiento de servidores:** si hay procesos corriendo en un servidor que necesita mantenimiento o una actualización, es posible migrar los procesos a otro servidor y proceder a desconectar del clúster al primero.

- **Priorización de trabajos:** en el caso de contar con varios procesos corriendo en el clúster, donde uno de ellos es de mayor importancia, este puede migrarse a los servidores que posean más o mejores recursos para acelerar su procesamiento.

5.4.5 Ambientes de Programación Paralela

Los ambientes de programación paralela permiten implementar algoritmos que hacen uso de recursos compartidos: CPU (*central processing unit*), memoria, datos y servicios.

5.4.5.1 MPI

El protocolo MPI es un sistema estandarizado y portable de paso de mensajes para su uso en una variedad de arquitecturas de computación paralela. El estándar define la sintaxis y la semántica de un núcleo de rutinas disponibles en una librería utilizada para escribir programas en C, C++ y Fortran.

6 Hardware Utilizado

6.1 Estaciones Meteorológicas Automáticas

Una Estación Meteorológica Automática (EMA) es un conjunto de dispositivos eléctricos y mecánicos que registran y transmiten mediciones de las variables meteorológicas de forma automática y numérica de los sitios donde se encuentran ubicados. [28]

Su función principal es el registro y monitoreo de variables meteorológicas (temperatura, humedad, velocidad y dirección del viento, presión atmosférica, lluvia y otros) que se almacenan en archivos a intervalos de tiempo regulares. Estos datos son enviados por una red celular, enlace microondas o vía satélite también a intervalos regulares o en tiempo real, la hora que se utiliza para registrar los datos es denominado horario UTC (Universal Time Coordinated).

Estas estaciones son versiones autónomas y automatizadas de las meteorológicas convencionales, que requieren de un operador humano para observar y registrar los datos. Las EMAs están preparadas para funcionar en forma autónoma las 24 horas del día sin necesidad de intervención o labor humana, lo cual es ideal para realizar mediciones en zonas remotas, inhóspitas o de difícil acceso. Es común verlas instaladas al lado de las estaciones convencionales.

Los datos registrados por una estación meteorológica instalada en un terreno que cumpla con todas las características ideales de entorno (planicie, sin bosques o edificios cercanos, distante de masas de agua, etc.), tienen una representatividad aproximada de cincuenta kilómetros a la redonda. [29]

En cuanto al uso de energía, normalmente funcionan con una batería que abastece a todo el conjunto de elementos de la estación, que es recargada por medio de energía solar o eólica.

6.1.1 Ventajas

- Adquisición de datos en tiempo variable de acuerdo a la necesidad del usuario.
- Posibilidad de colocar estaciones en lugares hostiles y de difícil acceso.

- Mayor precisión en los datos.
- Rapidez para realizar tareas estadísticas.

6.1.2 Desventajas

- Al contrario de las estaciones convencionales, las automáticas no pueden ofrecer reportes de clases ni cantidades de nubes.
- Las mediciones de lluvia (precipitación) son un poco problemáticas, ya que el medidor debe vaciarse por sí solo entre observaciones.
- Fenómenos que no se pueden medir por sensores, permanecen sin observación.
- Cuando un sensor se encuentra dañado o defectuoso, arroja lecturas erróneas en muchos casos difíciles de notar.

6.1.3 Sensores

Existe una gran variedad de sensores que pueden ser incorporados a una estación meteorológica automática, que van desde un termómetro (temperatura) hasta un medidor de muones (partículas de materia oscura). En este gran espectro de posibilidades, las mediciones más importantes para determinar el estado atmosférico en el entorno de la estación son realizadas con los siguientes sensores:

Termómetro: sensor capaz de captar la variación de temperatura y transformarlo en señales electrónicas. Pueden venir en diferentes formatos y modelos dependiendo del medio y uso requerido (aire, suelo, agua, gas, etc.)

Anemómetro: mide la velocidad del viento, existen diferentes formatos y modelos. Los más utilizados en las EMAS son los de tipo molinete que por medio de la fuerza del viento hace girar las aspas y la velocidad de ese giro es leída y transformada a señales electrónicas.

Higrómetro: instrumento para medir la humedad del aire, normalmente están constituidos por dos electrodos arrollados en forma de espiral, entre ellos se ubica un tejido impregnado en cloruro de litio acuoso. Si se aplica a estos electrodos una tensión alterna, el tejido se calienta y se evapora parte del contenido de agua. A una temperatura determinada, se consigue un equilibrio entre la evaporación por calentamiento el tejido y la absorción de agua de la humedad del ambiente por el cloruro de litio (material específicamente higroscópico). Analizando estos datos se puede definir con precisión el grado de humedad y transformarlo a señales digitales.

Pluviómetro: instrumento empleado para medir la precipitación del lugar en un tiempo determinado. El mecanismo consiste en un embudo que conduce el agua colectada a una pequeña cubeta triangular doble, con una bisagra en su punto medio.

Es un sistema cuyo equilibrio varía en función de la cantidad de agua en las cubetas. La inversión se produce generalmente a 0,2mm de precipitación, así que cada vez que caen 0,2mm de lluvia la báscula oscila, vaciando la cubeta llena, mientras comienza a llenarse la otra. Cada vez que la cubeta doble se mueve, es registrado el movimiento mediante sistemas electrónicos.

Barómetro: este instrumento se encarga de medir la presión atmosférica. Para lograr la medición utilizan un pequeño sensor de silicio, llamado Barocap, que es básicamente un diminuto aneroide mono cristalino sensible a la presión. El material utilizado para contenido de los barómetros digitales varía según la empresa que los fabrique y del modelo específico.

Veleta: es un sensor capaz de medir la dirección del viento. Es un dispositivo giratorio que consta de una placa vertical que gira libremente sobre un eje capaz de medir la posición en que se encuentra dispuesta la placa, de modo tal que al estar en presencia de un determinado viento, esta se coloque en dirección al mismo y se pueda realizar la lectura. Las veletas y anemómetros normalmente transmiten datos en tiempo real, debido a que existen fenómenos de muy corto plazo que necesitan ser registrados como por ejemplo las ráfagas de viento que ocurren en periodos que no superan los cinco minutos.

6.1.4 Almacenamiento de datos

Una EMA cuenta con un dispositivo denominado Data Logger, que permite registrar y almacenar las mediciones de los diferentes sensores que se le incorporan.

El Data Logger cuenta con un reloj interno que se encuentra configurado con horario UTC. Normalmente se encuentra configurado para que registre los datos de los sensores asociados en periodos de tiempos pre establecidos (cada 5, 10, 20 minutos, etc.). Los equipos profesionales ofrecen la posibilidad de introducir fórmulas y asociarlas a la lectura de uno o más sensores, de manera a que el valor arrojado sea el resultado del cálculo de la misma. Los datos son almacenados en una memoria local donde se registra el momento (fecha y hora) y los valores diversos obtenidos de cada sensor. Los registros existentes en la memoria son guardados en un archivo con formatos pre establecidos por la marca (MIS, CSV, ARK, etc.), inclusive algunos los almacenan en bases de datos portables como SQLite.

6.1.5 Transmisión de Datos

Los Data Loggers cuentan con varias opciones de conectividad como ser, puerto RS232, USB, LAN; inclusive algunos modelos vienen con un modem GSM integrado para conexión a una red celular. Adicionalmente se puede utilizar enlaces microondas

o módems satelitales, cualquiera de estas opciones es viable a la hora de transmitir datos. Normalmente la elección de algún tipo de conectividad depende de la ubicación geográfica de la estación meteorológica automática.

Los datos transmitidos por las EMAS son los archivos generados por los Data Loggers (Ver punto 6.1.4). En la mayoría de los casos, estos archivos representan las lecturas de un día; otros generan un archivo de datos para cada periodo de tiempo (Ej. Un archivo cada 10 minutos). En el caso de las bases de datos, los archivos son únicos y van creciendo en tamaño.

Dependiendo de la cantidad de estaciones existentes en la red, la transmisión de todos los datos se puede volver un problema. Para minimizar el tráfico, habitualmente se coloca un servidor de versionado a fin de transmitir solamente las modificaciones realizadas y no todo el contenido de cada archivo.

6.2 Estaciones Hidrológicas Automáticas

El funcionamiento de una Estación Hidrológica Automática (EHA) es similar al de una EMA, ambas se encargan de recolectar datos por medio de sensores asociados, almacenarlos en sus Data Loggers y transmitirlos a una central de almacenamiento. La diferencia radica en la ubicación de las estaciones y en los sensores asociados. Su situación varía debido a que se utilizan para realizar observaciones en cauces hídricos (ríos, arroyos, etc.).

Las EHAS habitualmente son instaladas a orillas de algún cauce hídrico; también, es común verlas sujetas a puentes o estructuras que atraviesan o circundan los mismos.

Las observaciones comúnmente asociadas a este tipo de estación son las de nivel, temperatura y conductividad del agua, aunque se acostumbra incorporar un pluviómetro para tener la cantidad de lluvia registrada en el lugar.

6.2.1 Sensores

Nivel de agua: es un sensor capaz de medir la altura de la columna de agua existente en un punto de un cauce hídrico.

Termómetro: se encarga de medir la temperatura del líquido.

Conductividad: realiza la medición del nivel de conductividad eléctrica del fluido.

6.3 Tarjetas de Memoria SD

Son dispositivos de almacenamiento de datos que utilizan memoria flash, comúnmente utilizados en dispositivos electrónicos como cámaras digitales, celulares, computadoras, notebooks, tabletas, reproductores MP3, SBCs y otros.

El concepto de memoria flash fue desarrollado por Toshiba, con una capacidad de borrar datos en un instante (“flash”). Estos chips son baratos para construir y proporcionan pequeñas densidades de bits.

6.3.1 Formatos de Tarjeta

Actualmente, el formato de tarjeta más común es el Secure Digital (o SD). También existen otros formatos como PC Card, Compact Flash I y II, Smart Media, Memory Stick y sus variantes, UFS, y otros.

6.3.2 Formato Secure Digital

El formato Secure Digital (SD) es un formato desarrollado en conjunto por SanDisk, Panasonic y Toshiba. Fue introducido en 1999 como una mejora evolutiva a las tarjetas MMC.

Esta tarjeta cuenta con cuatro versiones que a su vez están disponibles en tres tamaños. Las versiones son:

- Standard Capacity (SDSC), versión original, capacidad estándar
- High Capacity (SDHC), de alta capacidad
- Extended Capacity (SDXC), de capacidad extendida
- Input/Output (SDIO), entrada/salida

Los tres tamaños disponibles son: SD estándar original, miniSD y microSD.

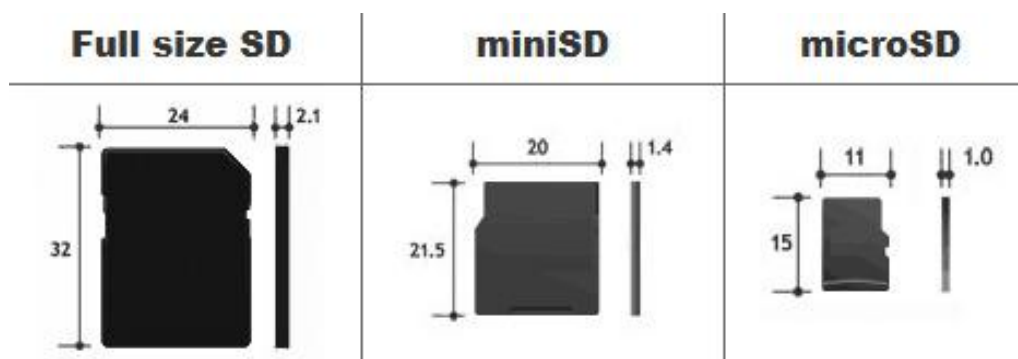


Ilustración 10: Tamaños de Tarjetas SD. Fuente: SD Association.

Las velocidades de transferencia mínimas actuales garantizadas para las tarjetas SD han sido estandarizadas como:

Speed Class	Velocidad mínima de escritura secuencial
Clase 2 (C2)	2MB/s
Clase 4 (C4)	4MB/s
Clase 6 (C6)	6MB/s
Clase 10 (C10)	10MB/s

Tabla 1: Velocidades de Tarjetas SD

6.4 Computadoras de Placa Reducida

Una SBC (Single Board Computer) es una computadora completa construida en una única placa de circuitos, con microprocesador, memoria, pines de entrada/salida (I/O) y otras características comúnmente requeridas por un ordenador funcional en una sola tarjeta que suele ser de tamaño reducido. Estos tipos de computadoras fueron diseñados para demostración y desarrollo de sistemas, para fines educativos o para su uso como computadora embebida en controladores en entornos industriales.

Debido al reducido tamaño de sus componentes y su capacidad de integración. Este tipo de computadoras suele ser más pequeña, liviana y con un requerimiento de energía eléctrica menor.

En comparación con una PC de escritorio, estas no se apoyan en bahías de expansión para soportar periféricos.

6.4.1 Tipos de SBC

Actualmente existen varias computadoras tipo SBC disponibles en el mercado, si bien todas ofrecen prestaciones similares, cada una cuenta con características resaltantes diferentes lo que origina la competencia por ser la placa más completa del mercado. Algunas de las marcas más renombradas en la actualidad son: Raspberry Pi, BeagleBoard, Banana Pi, Intel Edison, ODROID, Orange Pi, HummingBoard, Intel Nuk, pcDuino, entre otros.

6.4.2 Elección de un SBC

Debido a que la variedad de marcas y modelos de este tipo de computadores es muy amplia, se debe tener en cuenta los siguientes detalles:

Costo: el precio es siempre un factor importante, la mayoría de estas computadoras oscilan entre 9 y 99 dólares americanos.

Sistema operativo: normalmente este tipo de equipos cuenta con al menos un sistema operativo disponible para ser descargado, con todos los controladores específicos incorporados para cada modelo de tal forma que al descargarlo se encuentre listo para su uso. Los sistemas operativos más frecuentes son: Linux, Android y Windows. La elección podría depender del uso que se le dará al equipo.

Procesador: la gran mayoría de las SBCs cuentan con procesadores ARM de uno o varios núcleos, de 32 o 64 bits. Pero también existen en el mercado computadoras de este tipo con procesadores x86, como ser el Intel Nuk.

Memoria: si bien las SBCs cuentan con memoria integrada en la mayoría de los casos, también existen placas que cuentan con ranuras de memoria.

Características: existe una gran variedad de características en cada SBC, se pueden destacar: cantidad de puertos USB, salida de video, LAN, WiFi, Bluetooth, I/O, audio, etc. Accesorios compatibles como ser: cámara, pantallas, soportes, cajas o estuches, etc.

Documentación y Soporte: es importante a la hora de operar con estos equipos tener el respaldo de una comunidad con conocimientos que ayude a resolver los problemas que se puedan presentar y/o que ya se hayan presentado.

6.4.3 Selección del SBC para el proyecto

Para este proyecto de investigación se determinó que la SBC que mejor se adaptaba a las necesidades era el Raspberry Pi aunque contaba con dos desventajas; una es el procesador ARM, y la otra que su velocidad máxima a través de red LAN es de solamente 100Mbps.

6.4.4 Raspberry Pi

El Raspberry Pi es un SBC de muy bajo costo desarrollado por la Fundación Raspberry Pi. Fue diseñado y desarrollado con la intención de estimular la enseñanza de la informática y la electrónica en instituciones de enseñanza y países en desarrollo.

Varias versiones del Raspberry Pi fueron lanzadas de sus inicios en el 2012. La última versión lanzada es el Raspberry Pi 3 Modelo B (Ilustración 11).



Ilustración 11: Raspberry Pi 3 Modelo B. Fuente: Raspberry Pi Foundation.

Todos los modelos cuentan con un Sistema en Chip (System on a Chip o SoC) Broadcom, que incluye un CPU compatible ARM y un chip de procesamiento de gráficos o GPU. Las velocidades del CPU se encuentran en el rango de 700Mhz a 1.2Ghz y la memoria disponible de 256MB a 1GB. Tarjetas de tipo Secure Digital (SD) son utilizadas para almacenar el sistema operativo. Los modelos cuentan entre uno y cuatro puertos USB, salida HDMI y video compuesto (composite video) y una toma de audio de 3.5mm. Salida de bajo nivel están disponibles a través de un número de pines GPIO que soportan protocolos comunes como I2C.

En la Ilustración 12 se puede apreciar un diagrama de bloque que muestra la organización de los componentes internos del Raspberry Pi. Los Modelos A, A+ y el Pi Zero no cuentan con componentes USB ni Ethernet. Mientras que en los demás modelos, el adaptador de Ethernet se encuentra conectado a un puerto USB adicional, compartiendo de esta forma, un mismo bus de datos.

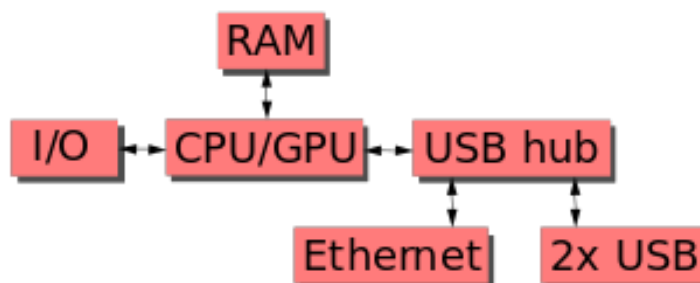


Ilustración 12: Componentes internos del Raspberry Pi.
Fuente: Raspberry Pi Foundation.

6.4.4.1 Requerimientos de Energía

El Raspberry trabaja con 5 voltios a través de una toma micro USB o por medio de conexión directa a los pines de I/O.

En la Tabla 2 se pueden observar los requerimientos de energía para cada modelo.

Modelo	Capacidad de corriente recomendada	Capacidad máxima por puerto USB	Consumo habitual en estado activo
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (conmutable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (conmutable)	~350mA
Raspberry Pi 3 Model B	2.5A	1.2A	~400mA

Tabla 2: Requerimientos de energía del Raspberry Pi

6.4.4.2 Almacenamiento

El formato de tarjeta soportado por el Raspberry Pi es el Secure Digital (SD). En las versiones más antiguas del SBC utilizaban el tamaño normal, pero las dos últimas versiones del equipo ya vienen preparadas para el tamaño más pequeño, el denominado Micro SD (Ver punto 6.3.2).

Para este proyecto se utilizaron dos tipos de tarjetas SD (Ver punto 6.5.2):

- MicroSD con capacidad SDHC y clase 10, para cada Nodo de Cómputo y para el Servidor Pos Procesamiento.
- MicroSD con capacidad SDXC y clase 10, para el Nodo de Acceso RPi.

6.5 Clúster de Raspberry Pi's

Como se ha mencionado en el punto 5.1, un clúster es un conjunto de computadoras unidas entre sí por una red dedicada de alta velocidad que tienen por objetivo trabajar

juntas como si fueran una sola súper computadora. Para este proyecto se construyó un clúster de Raspberry Pi's para emular el funcionamiento de uno de alto rendimiento.

6.5.1 Componentes principales del clúster

En la Tabla 3 se describen los principales componentes que integran el clúster de Raspberry Pi's.

Componente	Cantidad
Raspberry Pi 3 Modelo B	22
Tarjeta microSDHC 8GB C10	21
Tarjeta microSDXC 128GB C10	1
Rack cerrado de 6U	1
Fuente alimentación de 200 Watts	2
Ventilador disipador de calor	2
Switch de 24 puertos 10/100Mbps	1
Cable UTP Categoría 5	20 metros
Ficha RJ45	46
Ficha Dupont hembra 2,54mm 1x2P	22

Tabla 3: Componentes principales del clúster

6.5.2 Constitución del clúster

El clúster está conformado por un total de 22 Raspberry Pi's, de los cuales 20 son utilizados como nodos de cómputo, 1 como nodo de acceso y 1 como servidor pos procesamiento. El esquema conceptual del clúster propuesto se muestra en la Ilustración 13.

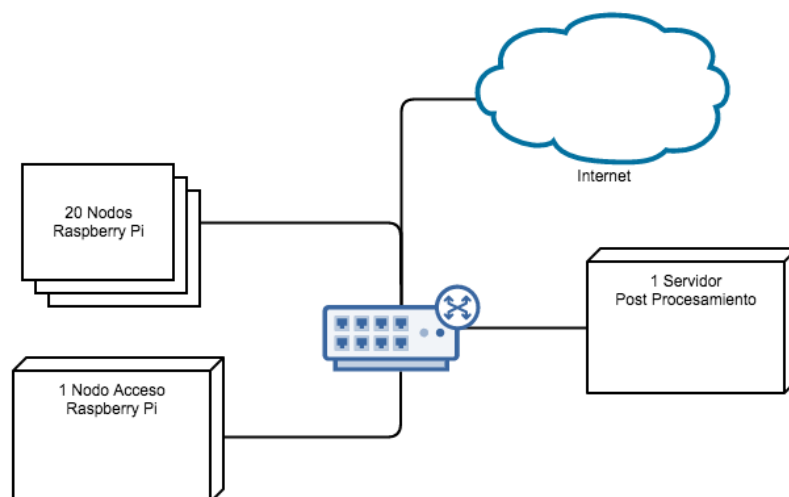


Ilustración 13: Esquema Conceptual del Clúster RPi. Fuente: Elaboración propia.

Todos los componentes del clúster están albergados en un Rack de 6U donde se encuentran distribuidos de forma ordenada y con una buena ventilación a fin de evitar el sobre calentamiento.

6.5.2.1 Disposición de Raspberry Pi's

Para incorporar los RPi's a la estructura del Rack se fabricó una bandeja especial donde se encuentran dispuestos en forma de seis torres de equipos. Los nodos se encuentran situados en cinco torres de cuatro RPi's cada una; mientras que el Nodo de Acceso y el de Pos Procesamiento hacen juntos una única torre de dos elementos. Esta configuración permite una mejor circulación de aire y una fácil disposición de los cables de red y alimentación. Ver Ilustración 14.



Ilustración 14: Disposición de RPi's. Fuente: Elaboración propia.

6.5.2.2 Conexión de Red

Tanto el Nodo de Acceso como los nodos de procesamiento y el servidor de Pos Procesamiento se encuentran conectados al Switch, por medio del cual logran la conectividad necesaria para comunicarse entre sí. A su vez, se encuentra conectado a una red externa de la que obtiene acceso a internet, tal como se puede observar en la Ilustración 15.

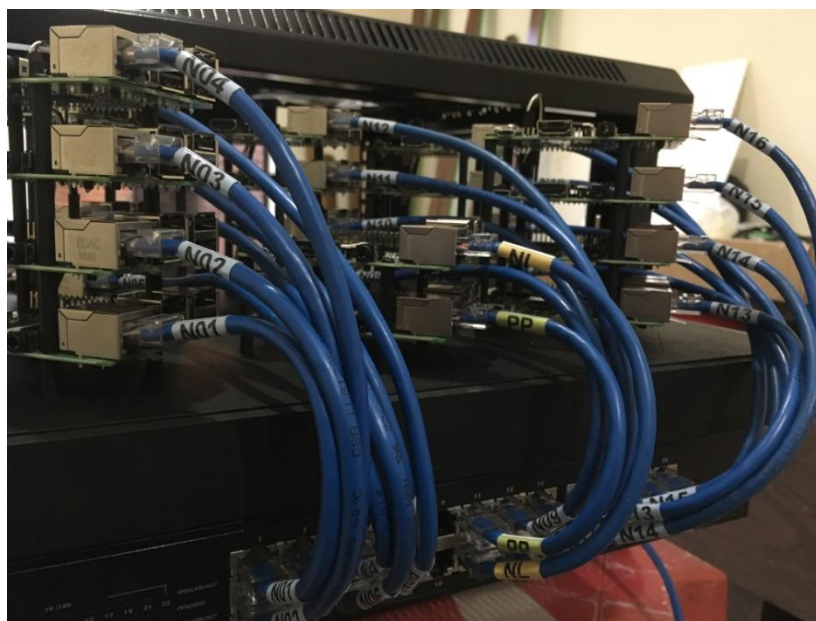


Ilustración 15: Conexión de Red entre RPi's. Fuente: Elaboración propia.

6.5.2.3 Provisión de Energía

La alimentación de energía para el equipo se realiza por medio de un cable de conexión que debe estar conectado a una corriente alterna de 220 Voltios, esta al ingresar pasa directamente por una llave térmica de 10A y luego es distribuida a las fuentes y al Switch. Todos los componentes del clúster, incluso el rack, cuentan con salida a tierra.

Para una distribución de energía equilibrada, se realizó una agrupación de las Torres de RPi's; quedando divididos en dos conjuntos de torres, uno con tres torres de cuatro RPi's (doce equipos), y el otro con dos torres de cuatro y una de dos (diez equipos). Cada conjunto de torres es alimentado por una fuente de alimentación de 5V.

En la parte inferior del Rack, se instalaron las dos fuentes de alimentación de 5V que proveen de energía a los conjuntos de torres de Raspberry Pi's (tres torres cada una), y una fuente de 12V que alimenta a los extractores de aire, como se muestra en la Ilustración 16.



Ilustración 16: Alimentación Cluster. Fuente: Elaboración propia.

Tal como se especifica en el punto 6.4.4.1, los Raspberry Pi's se alimentan mediante una corriente continua de 5V y tienen un consumo de 2.5A cada uno para el correcto funcionamiento de todos los elementos, serán necesarios 295 Watts (150 Watts para el conjunto de torres con doce elementos, y 145 Watts para el otro conjunto de diez elementos). Las fuentes de 5V utilizadas son de 200 Watts cada una, con lo que se obtuvo una salida equilibrada de 40A para cada conjunto de torres de RPi's. En cuanto a la alimentación de energía de los extractores de aire, se utilizó una fuente de 12V de 5A.



Ilustración 17: Alimentación RPi's.
Fuente: Elaboración propia.

Para conectar cada fuente a la corriente alterna se utilizó cables de 2mm, mientras que para llevar la corriente continua a los grupos de torres de RPi's y a los extractores, se utilizó cables de 1mm. La alimentación individual de cada Raspberry Pi se realizó con cables de 0,25mm y fichas Dupont hembras como terminales (ver Ilustración 17 e Ilustración 18).

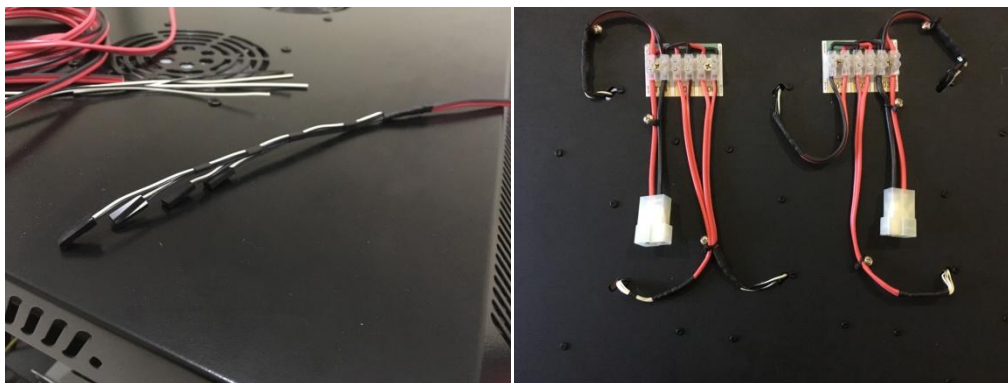


Ilustración 18: Cables de Alimentación. Fuente: Elaboración propia.

6.6 Servidor de Pruebas

En el marco de la implementación del presente proyecto, la Dirección de Meteorología e Hidrología proporcionó un servidor para realizar las pruebas correspondientes. Se trata de un equipo que cuenta con las siguientes características:

- 4 Procesadores Intel Xeon de 2.4Ghz con 10 núcleos cada uno (total 40 núcleos).
- Memoria RAM de 64GB.
- Disco Duro 1.6TB.

El servidor se encuentra físicamente alojado en el datacenter de la institución y permite el acceso remoto vía SSH.

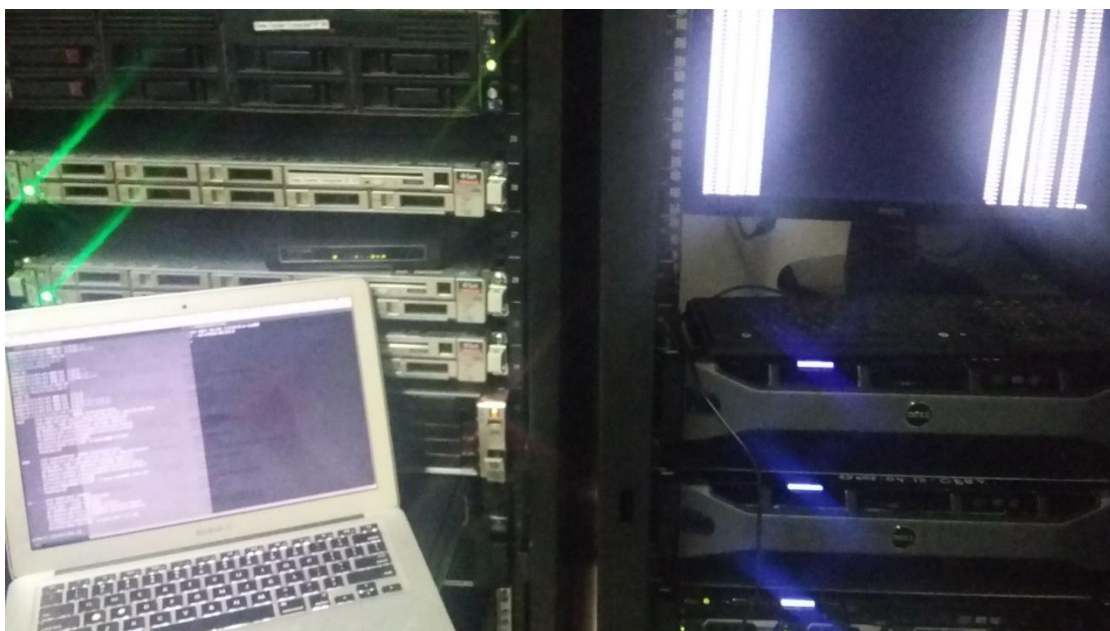


Ilustración 19: Datacenter DMH - Servidor de Pruebas. Fuente: Elaboración propia.

7 Software utilizado

Para la elaboración de los diferentes productos resultantes del proyecto, se utilizó una gran variedad de herramientas de software como ser: sistemas operativos, lenguajes de programación, bases de datos, gestores de mensajes, frameworks, utilitarios y diversas librerías.

7.1 Sistemas Operativos

Un Sistema Operativo (por sus siglas SO) es el software principal que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software, ejecutándose en modo privilegiado respecto de los restantes. En la actualidad los SO más utilizados son: Windows, Linux, macOS y Android, todos ellos cuentan con varias distribuciones y versiones que fueron evolucionando a través del tiempo.

7.1.1 Elección del Sistema Operativo

Linux es un SO robusto y seguro; capaz de manejar grandes cantidades de procesos. Además es gratuito y de código abierto.

Las distribuciones Linux usan repositorios oficiales para sus aplicaciones, aplicando controles de calidad y políticas de seguridad que garantizan la integridad y sanidad de los paquetes instalados.

Además de ser un Sistema Operativo estable y seguro, todas las aplicaciones requeridas, se ejecutan en Linux, por estas razones se optó por elegirlo como el sistema operativo principal para el desarrollo del proyecto.

7.1.1.1 Sistema Operativo del Servidor de Pruebas

Para el servidor de pruebas se utilizó el Sistema Operativo Ubuntu Server en su versión 16.04 que cuenta con soporte a largo plazo (por sus siglas en inglés LTS).

Ubuntu es un sistema operativo basado en GNU/Linux, que se distribuye como software libre. Su patrocinador, Canonical, es una compañía británica que ofrece el

sistema de manera gratuita y se financia por medio de servicios vinculados al sistema operativo y vendiendo soporte técnico.

7.1.1.2 Sistema Operativo del Clúster de RPi's

Como se menciona en el punto 6.4.4, los Raspberry Pi's son desarrollados por la Fundación Raspberry Pi, la misma a su vez provee un sistema operativo para sus equipos denominado Raspbian, se trata de una distribución de Linux basada en Debian. Además, pone a disposición en su página web otras versiones que funcionan con el RPi como ser: Ubuntu, Windows 10 IOT Core, RISC OS y distribuciones especializadas para centros de medio.

En este proyecto se optó por utilizar el Sistema Operativo Raspbian, debido a que es un SO maduro con mucha documentación y soportado por la Fundación Raspberry Pi.

Cada elemento del clúster tiene instalado el Sistema Operativo Raspbian en su versión Jessie Lite que se encuentra basado en Debian Jessie.

7.2 Lenguajes de programación

Para el desarrollo de este proyecto, fue necesario utilizar varios lenguajes de programación, frameworks y librerías. La elección de los lenguajes utilizados estuvo basada en la disponibilidad de frameworks como en las librerías disponibles para realizar las distintas tareas de investigación.

7.2.1 PHP

El lenguaje principal utilizado en este trabajo de investigación fue PHP, que son las siglas de "PHP: Hypertext Processor". Creado por Rasmus Lerdorf en 1994, es un lenguaje de tipo script interpretado con HTML embebido. Gran parte de su sintaxis es similar a C, Java y Perl con algunas de características únicas del lenguaje. El objetivo del lenguaje es el de permitir a programadores escribir aplicaciones web que permitan generar páginas HTML de forma rápida.

El intérprete estándar de PHP, el Zend Engine, es software libre publicado bajo la licencia PHP License. PHP ha sido portado a muchas arquitecturas y puede ser utilizado en la mayoría de los servidores web en casi cualquier sistema operativo, sin costo de licencias. [30]

7.2.2 JavaScript

Es un lenguaje de programación interpretado, asíncrono, dinámico, orientado a objetos, basado en prototipos, imperativo y de alto nivel.

Se utiliza principalmente en su forma del lado del cliente (Client-Side), implementado como parte del navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-Side JavaScript o SSJS). Es también significativo su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets).

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

JavaScript se ha convertido en uno de los lenguajes de programación más populares en internet, todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web, como resultado de esto hubo una proliferación de un conjunto de frameworks y librerías de ámbito general, mejorando las prácticas de programación con JavaScript.

7.2.3 Python

Python es un lenguaje de programación de alto nivel, multi propósito, creado por Guido van Rossum a comienzos de 1990 [31]. La filosofía del diseño del lenguaje enfatiza la legibilidad del código y una sintaxis que permite a los programadores expresar conceptos en menos líneas de código en comparación con lenguajes como C++ o Java. Python al igual que PHP es un lenguaje interpretado.

Existen intérpretes Python disponibles para muchos sistemas operativos lo que permite correr código Python en una gran variedad de sistemas.

7.2.4 NCL

NCL son las siglas de "NCAR Command Language", un lenguaje interpretado diseñado específicamente para el análisis y visualización de datos científicos [32]. El lenguaje fue diseñado y creado por la National Center and Atmospheric Research con fondos proveídos por la National Science Foundation del Gobierno de los Estados Unidos.

Existen intérpretes de NCL disponibles para las plataformas Linux (CentOS, Debian, RHEL y SUSE) y Mac OS (OSX y macOS).

7.3 Bases de Datos

7.3.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto publicado bajo la licencia GPL2. Creado originalmente por la empresa MySQL AB. Es considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. Está escrita principalmente en C y C++, mientras que su parser SQL está escrito en yacc.

Está disponible en una gran cantidad de plataformas y es el componente central del grupo de tecnologías de desarrollo conocido como LAMP (Linux-Apache-MySQL-PHP) y WAMP (Windows-Apache-MySQL-PHP).

7.3.2 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiar a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará a los demás y el sistema continuará funcionando. [33]

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

7.4 Cola de Mensajes

7.4.1 Beanstalk

Beanstalk es un servidor de cola de mensajes rápido y simple. Fue creado inicialmente por Philotic, Inc. para reducir el tiempo de respuesta de su aplicación "Causes on Facebook", que en su momento tuvo 9.5 millones de usuarios. [34]

El servidor implementa un protocolo de comunicación exclusivo con el cliente para lograr optimizar el tiempo de respuesta.

7.5 Frameworks

7.5.1 Laravel

Es un framework de código abierto publicado bajo la licencia MIT, creado para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".

Fue creado por Taylor Otwell y publicado por primera vez en Junio de 2011. Laravel implementa el patrón MVC (Model-View-Controller) o Modelo-Vista-Controlador. Tiene una gran influencia de frameworks como Symfony, Ruby on Rails, Sinatra y ASP.NET MVC.

Una de las características principales del framework es que implementa un sistema de paquetes modular con un gestor dedicado de dependencias y soporta el acceso a distintos tipos de bases de datos relacionales.

7.5.2 Vue.js

Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue fue diseñado desde el inicio para ser incrementalmente adaptable. La biblioteca principal se centra únicamente en la capa de vista y es muy fácil de captar e integrar con otras bibliotecas o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de sustentar sofisticadas aplicaciones de una sola página (Single-Page) cuando se utiliza en combinación con herramientas modernas y bibliotecas de apoyo.

7.5.3 AngularJS

AngularJS (comunalmente llamado "Angular" o "Angular.js"), es un framework de Javascript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página (Single-Page). Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

AngularJS está construido en torno a la creencia de que la programación declarativa es la que debe utilizarse para generar interfaces de usuario y enlazar componentes de software, mientras que la programación imperativa es excelente para expresar la lógica de negocio. Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data binding bidireccional que permite la sincronización automática de modelos y vistas. Como resultado, AngularJS pone menos énfasis en la manipulación del DOM y mejora la testeabilidad y el rendimiento.

7.5.4 Ionic

Ionic es un framework de aplicaciones móviles HTML5 orientado a la creación de aplicaciones móviles híbridas. Las aplicaciones híbridas son esencialmente pequeños sitios web que se ejecutan en una terminal del navegador en una aplicación que tiene acceso a la capa de plataforma nativa. Las aplicaciones híbridas tienen muchos beneficios sobre aplicaciones nativas puras, específicamente en términos de soporte de plataforma, velocidad de desarrollo y acceso a código de terceros. [35]

7.6 Librerías

7.6.1 Laravel Collective SSH

El paquete SSH de Laravel Collective es un paquete para Laravel que provee una interfaz para ejecución de comandos SSH en servidores remotos. Ésta librería permite la gestión de varios servidores con sus credenciales SSH, así como la ejecución de uno o más comandos SSH en estos servidores.

Adicionalmente, permite definir bloques de comandos SSH como "tareas" que luego pueden ser ejecutadas. También permite transferir o descargar archivos desde los servidores remotos como así también, capturar la salida de los comandos ejecutados.

7.6.2 Pheanstalk

La librería Pheanstalk es un cliente PHP para acceso a servidores Beanstalk. Laravel soporta por defecto Beanstalk dentro de su componente Queues.

7.6.3 Guzzle

Guzzle es un cliente HTTP para PHP. Ofrece una interfaz de programación que permite de forma muy sencilla enviar peticiones HTTP y es muy utilizado como base para la integración de servicios web.

Ofrece una interfaz simple para crear peticiones de tipo GET y POST, cargar o descargar datos de gran tamaño y también permite enviar peticiones síncronas o asíncronas.

7.6.4 NumPy

NumPy es una librería de Python que ofrece utilidades para la computación científica como así también, para el manejo de datos científicos.

La librería es de código abierto y está publicada bajo la licencia BSD.

7.6.5 netCDF4 module

NetCDF (Network Common Data Form) es un conjunto de librerías y formatos de datos auto descriptivos e independientes a la plataforma, capaz de crear, acceder y compartir colecciones de datos científicos (array-oriented scientific data).

HDF (Hierarchical Data Format) o Formato de Datos Jerárquicos, es un conjunto de formatos de archivos (HDF4 y HDF5) diseñado para almacenar y organizar grandes volúmenes de datos.

La denominada netcdf4-python, es una librería para Python que provee interfaz a la librería netCDF escrita en C.

La versión 4 del netCDF tiene muchas características que no están disponibles en versiones anteriores de esta librería y está implementada sobre HDF5. Esta librería permite leer y escribir archivos netCDF, y puede crear archivos que son accesibles para clientes HDF5.

7.6.6 MPICH

La implementación de MPI utilizada por el Modelo WRF es el MPICH, el cual es una implementación open source desarrollada por el Argonne National Laboratory de los Estados Unidos. Está disponible para Linux y OS X.

7.7 Utilitarios

7.7.1 Win32DiskImager

Es un utilitario que sirve para grabar archivos de imagen de disco IMG en tarjetas SD y memorias USB. También funciona a la inversa ya que es capaz de hacer copias de seguridad de los mismos dispositivos.

7.7.2 Ansible

Es una herramienta de automatización open source diseñada y desarrollada para la gestión de configuración, la provisión de servidores en la nube e implementación de aplicaciones.

Ansible gestiona los nodos a través del protocolo SSH y no requiere de ningún software adicional para funcionar (arquitectura sin agentes). Se distinguen dos tipos de

servidores, controladores y nodos. El servidor controlador es el que controla a los nodos y la ubicación de los nodos se describe en un archivo de inventario.

Para gestionar todos los nodos, se ejecutan los comandos en el servidor controlador, Ansible luego se conecta vía SSH a cada nodo para ejecutar el comando. De esta forma, Ansible no necesita un agente corriendo en los nodos en segundo plano.

8 Configuraciones, parametrizaciones, ajustes y validación del Modelo WRF

En este capítulo se describen los ajustes realizados a los componentes del WRF para compilarlos y ejecutarlos en una arquitectura ARM. Asimismo, se define el dominio a ser utilizado para la simulación y las parametrizaciones a ser configuradas en el mismo. Por último, se detallan la configuración del WRF para su ejecución en el Clúster y la validación de las salidas del modelo.

8.1 Compilación del Modelo WRF para la Arquitectura ARM

Para correr el WRF en el Clúster, es necesario configurar y compilar el modelo para la arquitectura ARM.

Escrito en Fortran, el WRF requiere de un compilador Fortran compatible para generar los ejecutables del modelo.

La versión más reciente del WRF, la versión 3.8.1, publicada en agosto de 2016 no incluye soporte por defecto para esta arquitectura.

Los compiladores requeridos para generar los binarios del modelo son el *gfortran*, el *cpp* y el *gcc*. Todos estos compiladores están disponibles en el Sistema Operativo Raspbian del Raspberry Pi.

Además de los compiladores, el WRF requiere de las siguientes librerías que deben estar instaladas en el Sistema Operativo: *MPICH 3.0.4*, *NetCDF 4.1.3*, *Jasper 1.900.1*, *libPNG 1.2.50*, *zlib 1.2.7*. [36]

En el punto 10.2 se presentan las pruebas realizadas para lograr compilar y generar los binarios para la arquitectura ARM.

8.1.1 Preparación de Imágenes para los Nodos del Clúster

Se creó una versión genérica de uno a la que se denominó Nodo Genérico, para evitar instalar y configurar los nodos de cómputo individualmente.

El nodo genérico fue instalado y configurado con los componentes necesarios para el correcto funcionamiento del modelo y se le asignó una dirección IP fija en la red para que no tenga conflicto con los demás componentes del clúster.

Una vez preparado el nodo genérico, se procedió a realizar una copia de seguridad de la tarjeta SD que lo contiene utilizando la herramienta *Win32DiskImager*.

Las tarjetas SD para los 20 nodos de cómputos fueron creadas a partir de la imagen generada con la copia de seguridad del nodo genérico. El proceso de grabar la imagen en una tarjeta SD demoró aproximadamente 25 minutos.

Con las tarjetas grabadas, se procedió a asignar a cada nodo de cómputo una dirección IP fija dentro de la red, esta tarea se debió realizar nodo a nodo, debido a que todas tienen el mismo IP y crearía conflicto en la red.

8.2 Configuración del dominio

Para generar los pronósticos meteorológicos de la ciudad de Encarnación se definió un dominio con una resolución de 5 km, en donde la ciudad se encuentra ubicada en el centro del mismo (ver Ilustración 20).

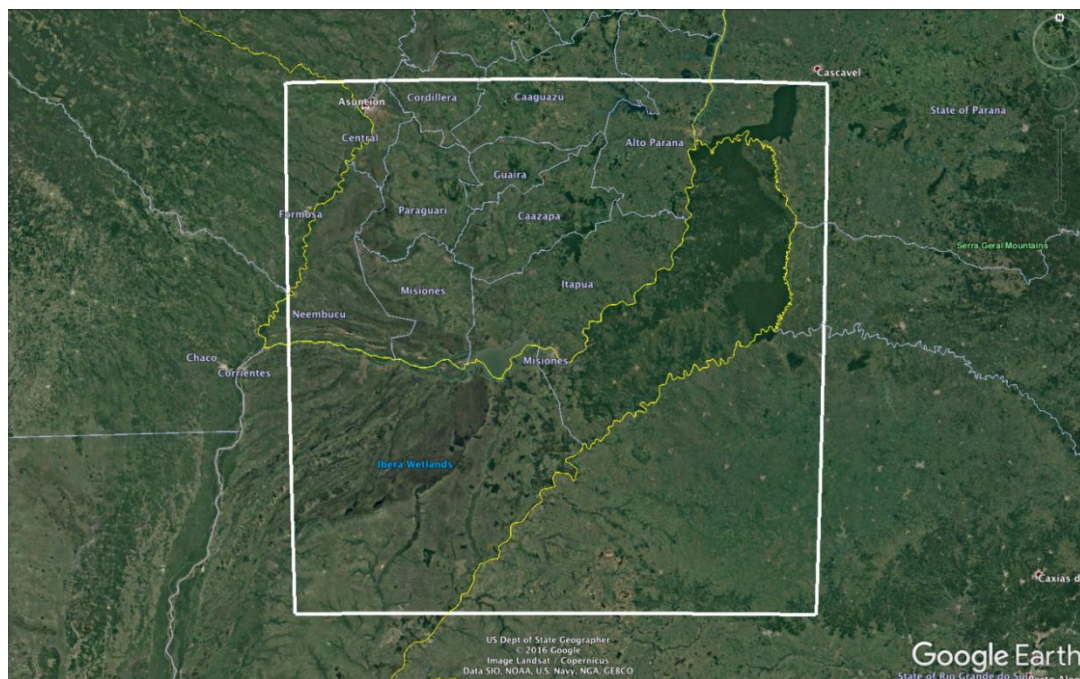


Ilustración 20: Configuración del dominio centrado en Encarnación. Fuente: Elaboración propia.

El tamaño de cada grilla dentro del dominio es de 5 por 5 kilómetros. Se estableció un tamaño del dominio de 100 grillas horizontales por 100 grillas verticales con la intención de abarcar la mitad de la región oriental; de esta forma, se podrá comparar los pronósticos generados por el modelo con alrededor de 15 EMAS que forman parte de la red de estaciones de la DMH (ver Ilustración 21). Gracias a ello se podrá generar pronóstico para todos distritos de Itapúa.

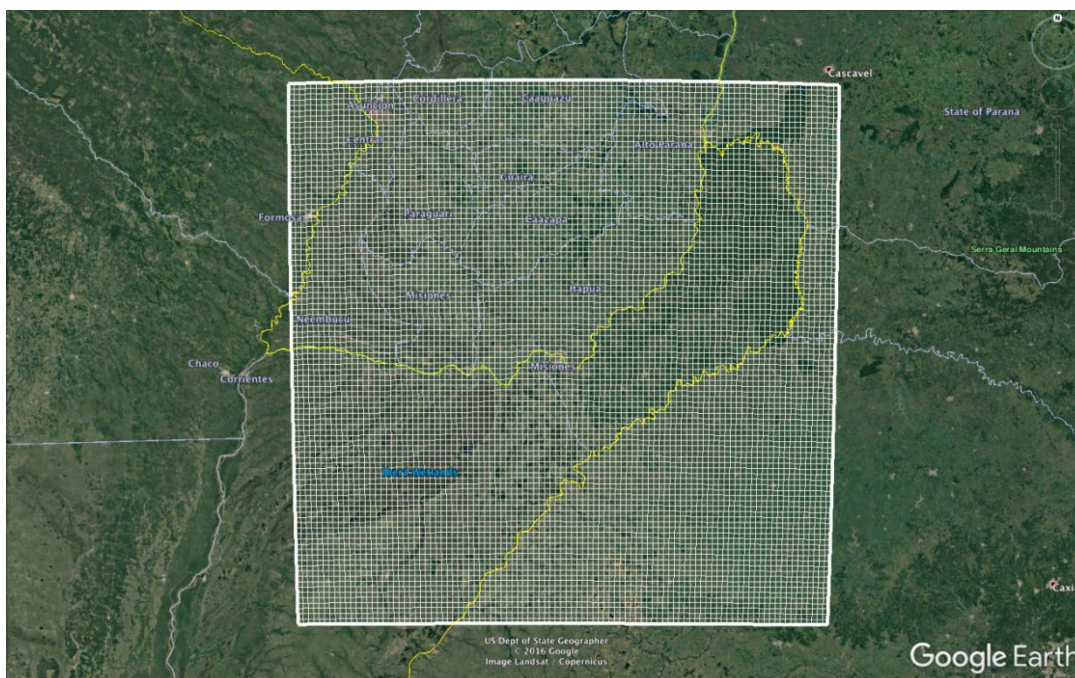


Ilustración 21: Grillas del dominio configurado. Fuente: Elaboración propia.

8.3 Parametrización del Dominio

En el punto 10.4.1 se detallan las pruebas realizadas para obtener la parametrización adecuada del dominio.

Los parámetros finales de configuración del dominio se muestran en la Tabla 4.

Parámetro	Descripción	Valor
Parametrización del Dominio		
time_step	Time step for integration seconds	30
e_vert	End index in z (vertical) direction	40
p_top_requested	Pressure top (in Pa) to use in the model	5000
num_metgrid_levels	Number of vertical levels in WPS output	32

num_metgrid_soil_levels	Number of soil levels or layers in WPS	4
Parametrización de la Física para el Dominio		
mp_physics	Microphysics Options	Thompson graupel scheme
ra_lw_physics	Longwave Radiation Option	Rrtmg scheme
ra_sw_physics	Shortwave Radiation Option	Rrtmg scheme
radt	Minutes between radiation physics calls	10
sf_sfclay_physics	Surface Layer Option	Monin-Obukhov (Janjic Eta) scheme
sf_surface_physics	Land-surface Option	unified Noah land-surface model
bl_pbl_physics	Boundary Layer Option	Mellor-Yamada-Janjic (Eta) TKE scheme
cu_physics	Cumulus Parameterization Option	No cumulus parameterization
num_soil_layers	Number of soil layers in land surface model	4
sf_urban_physics	Urban Physics Option	Off

Tabla 4: Parametrizaciones del Dominio

8.4 Configuración del WRF en el Clúster

La ejecución del modelo en un clúster requiere que; por una parte, la librería MPI esté instalada en el Sistema Operativo, y por la otra que una copia del archivo binario a ser ejecutado esté disponible en cada nodo de cómputo.

En el punto 10.2.2 se mencionan las pruebas realizadas para lograr compilar satisfactoriamente el modelo en la plataforma ARM y así crear los archivos binarios necesarios para la ejecución del modelo.

Debido a que los ejecutables deben estar en cada nodo, fue indispensable buscar una manera de compartir fácilmente los archivos binarios del modelo con los nodos de cómputo del clúster.

Para compartir la carpeta del WPS y del WRF situada en el nodo de acceso con los demás se utilizó NFS. Para esto se creó una carpeta *wrf* en la raíz del sistema operativo ("/wrf") del nodo de acceso, se configuró y se exportó como compartida.

Los nodos de cómputo a su vez, montan la carpeta *wrf* compartida en su sistema, logrando de esta forma distribuir automáticamente los archivos binarios del modelo en todos los nodos.

En el punto 10.3.2 se describen los ensayos realizados para compartir los archivos binarios con todos los nodos de cómputo del clúster.

8.5 Validación del WRF

Fue necesario realizar una validación para verificar que el pronóstico generado por el modelo se aproxima a la realidad.

Se definieron cinco variables por ser las más significativas: temperatura a 2 metros del suelo; presión atmosférica a nivel del mar; dirección y velocidad del Viento a 10 metros del suelo; y precipitación.

Para la validación del modelo se propuso realizar la simulación para 31 días utilizando los archivos históricos de las salidas del modelo global GFS existentes en la DMH. La institución cuenta con estos archivos debido a que son utilizados por sus pronosticadores (Ver punto 10.6).

Se efectuó un análisis de datos disponibles, seleccionando el periodo comprendido entre el 3 de noviembre de 2016 al 3 de diciembre de 2016. Los mismos corresponden al ciclo de las 18 UTC y cuentan con 78 horas de simulación por día. En base a esto, se decidió generar los pronósticos a 72 horas o 3 días.

Adicionalmente, se obtuvieron datos observados de las EMAS del aeropuerto de Encarnación, de la Facultad de Agronomía de Hohenau y de la base aérea de Capitán Meza, todas de la red de estaciones de la DMH.

En el punto 10.6 se muestran las pruebas realizadas para generar los pronósticos y las validaciones realizadas, contrastando los resultados obtenidos con las observaciones efectuadas en las EMAS.

En el punto 12.4 del anexo se adjunta la nota remitida por la DMH donde informan que los pronósticos generados por el modelo propuesto para el dominio configurado son válidos.

9 Sistemas desarrollados

Para el presente trabajo de investigación, se requirió el desarrollo de sistemas a fin de alcanzar los objetivos específicos propuestos: Sistema de Gestión de datos de Estaciones Meteorológicas e Hidrológicas Automáticas (GEMHA), Sistema de Configuración, Automatización y Monitoreo del Modelo WRF (WRF Runner), y una aplicación móvil denominada *El Tiempo en Encarnación*. En este capítulo, se describen los sistemas que fueron desarrollados.

9.1 GEMHA

La Dirección Nacional de Meteorología e Hidrología cuenta con datos generados por las Estaciones Meteorológicas Automáticas (EMAS); sin embargo, no poseen una API de acceso a estos datos.

En visitas a la DMH, se pudo conocer el funcionamiento actual de la red de estaciones y la infraestructura de IT de la misma. Es importante destacar que la institución otorgó a los tesisistas credenciales con permisos suficientes para acceder a los datos de la red de estaciones automáticas.

Esta red posee además estaciones hidrológicas automáticas (EHAS), si bien no forma parte del trabajo de tesis, se hace referencia en este capítulo debido a que forma parte del Sistema GEMHA.

El sistema de Gestión de datos de Estaciones Meteorológicas e Hidrológicas Automáticas (GEMHA) procesa los datos generados por las estaciones, brinda un acceso programático a estos datos a través de una API y una interface administrativa que permite administrar el sistema.

9.1.1 Antecedentes y análisis de la situación actual

Se realizó un relevamiento de la situación actual de la red de estaciones meteorológicas e hidrológicas automáticas de la DMH.

Esta consta de un total de 64 estaciones meteorológicas automáticas; de las cuales, 41 son propiedad de la DMH, y 23 son de la Federación de Cooperativas de Producción

(FECOPROD) mantenidas por la DMH. Forman parte de esta red, 12 estaciones hidrológicas automáticas que son propiedad de la Secretaría del Ambiente (SEAM) que también reciben mantenimiento de la DMH (ver punto 12.1).

Como se menciona en el punto 6.1.4, las EMAS de esta red también almacenan las observaciones de sus sensores en distintos formatos; y la transmisión de datos (ver punto 6.1.5) es realizada cada 10 minutos a un servidor FreeNAS ubicado en el Data Center de la DMH.

Este servidor FreeNAS a su vez, publica carpetas compartidas como *solo lectura* que pueden ser montadas en otros servidores para su procesamiento.

9.1.2 Detalles del sistema

La aplicación recolecta, procesa y valida los datos reportados por los sensores de las estaciones meteorológicas e hidrológicas automáticas de la red de estaciones de la DMH.

Este sistema tiene las siguientes funciones:

9.1.2.1 Recolección de datos

Se recolectan los datos de cada estación meteorológica e hidrológica automática a través de un proceso interno que recopila la información almacenada en una carpeta específica asignada a cada estación, organiza los archivos y/o datos disponibles y los ordena para su posterior procesamiento.

9.1.2.2 Procesamiento de datos

Se procesan los datos de cada estación para convertirlos a un formato estándar. Para esto, se implementaron analizadores sintácticos específicos para cada tipo de estación de acuerdo al formato de datos que utiliza para registrar las observaciones de cada uno de sus sensores.

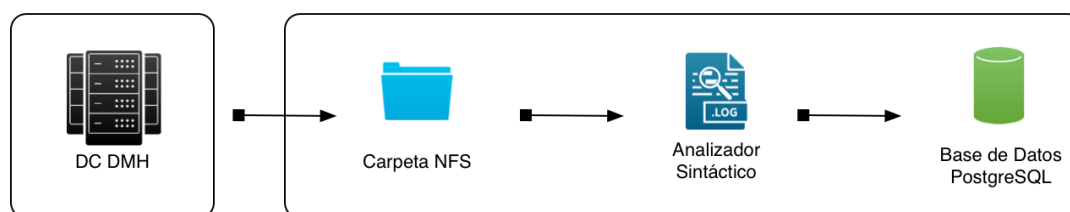


Ilustración 22: GEMHA - Procesamiento de datos. Fuente: Elaboración propia.

9.1.2.3 Validación de datos

Se validan los datos obtenidos para verificar que se encuentren dentro de rangos aceptables para cada tipo de medición leída (temperatura, humedad, presión, etc.). Los datos que estuvieran fuera de rango se marcan como no válidos, posteriormente pueden ser corregidos de forma manual a través de la aplicación por usuarios autorizados.

9.1.2.4 Estandarización de Datos

Se definió un formato estándar de almacenamiento de los datos de las estaciones.

Para el almacenamiento de los valores de los sensores se estableció como estándar al Sistema Métrico; mientras que, para los valores de tipo fecha y hora, la zona horaria UTC/GMT.

9.1.2.5 Visualización de datos

Mediante una página web se pueden visualizar los datos actuales de las estaciones meteorológicas e hidrológicas automáticas distribuidas en un mapa. Además se incorpora una opción para cada estación denominada: “ver datos”, la cual permite ingresar a la página de la estación donde se muestran los metadatos de la estación, últimos datos observados y datos históricos en formato tabular y de gráficos (ver punto 9.1.3.2).

9.1.2.6 Acceso estandarizado a datos

Se disponibiliza el acceso a los registros de la aplicación a través de una API siguiendo la arquitectura REST (Representational State Transfer) y utilizando como transporte de datos el formato JSON (Javascript Object Notation). Los parámetros requeridos para acceder a ciertos recursos deberán enviarse por el método GET.

Esta interfaz pone a disposición los siguientes recursos:

- **Estaciones meteorológicas:** mediante este recurso se pueden obtener un listado de las estaciones meteorológicas activas registradas en el sistema junto con sus metadatos.
- **Estaciones hidrológicas:** mediante este recurso se pueden obtener un listado de las estaciones hidrológicas activas registradas en el sistema junto con sus metadatos.
- **Detalles de estación meteorológica:** para obtener datos de este recurso se debe especificar la estación requerida pasando como parámetro el dato *station_id* (ID

de la estación). La aplicación devuelve un arreglo con los metadatos de la estación y los datos de la última observación registrada.

- **Detalles de estación hidrológica:** para obtener datos de este recurso se debe especificar la estación requerida pasando como parámetros el dato *station_id* (ID de la estación). La aplicación devuelve un arreglo con los metadatos de la estación y los datos de la última observación registrada.
- **Observaciones:** este recurso retorna los datos históricos registrados por una estación en el rango de fechas establecido. Para acceder a los mismos se deben proveer los siguientes parámetros: *station_id* (ID de la estación), *start_date* (Fecha inicial), *end_date* (Fecha final), *fields* (Campos a retornar).

9.1.3 Componentes del sistema

Este sistema se estructuró en varios componentes, permitiendo su fácil manutención y escalabilidad.

9.1.3.1 Componente backend o administrador

El componente backend está compuesto de varios sub componentes que permiten administrar la aplicación, recolectar y procesar los datos:

Comando de recolección de datos

Es una instrucción que chequea si existen datos a procesar de las estaciones. En caso que el resultado sea positivo, se dispara un "Job" (o Tarea) que procesa los datos.

Job (Tarea) de procesamiento de datos

Se encarga de analizar los metadatos de la estación para determinar dónde se encuentran almacenados sus archivos correspondientes y definir el analizador a ser utilizado (de acuerdo al tipo de estación) para obtener los nuevos datos. Una vez definidos estos parámetros, se ejecuta el analizador sintáctico.

Analizador sintáctico de datos

Se encarga de interpretar los archivos de la estación, verificar los datos, validarlos, y; por último, estandarizarlos. Adicionalmente, este sub componente se encarga de guardar los datos procesados en la base de datos.

Proceso de recolección y procesamiento de datos

Se encarga de ejecutar el comando de recolección de datos a intervalos regulares (actualmente cada 10 minutos). Por cada estación activa del sistema se crea una tarea de procesamiento de datos, la cual se coloca en una cola de procesamiento.

Un worker (o procesador de tarea), es el encargado de verificar si alguna tarea de procesamiento de datos ha sido agregada a la cola, de ser así, toma esa tarea y empieza a procesarla.

Se pueden tener varios workers en ejecución continua (actualmente se encuentran configurados ocho), de esta forma es posible procesar los datos de varias estaciones en simultáneo optimizando el tiempo de procesamiento de los datos de todas las estaciones.

El proceso de recolección y procesamiento de datos se muestra en la Ilustración 23

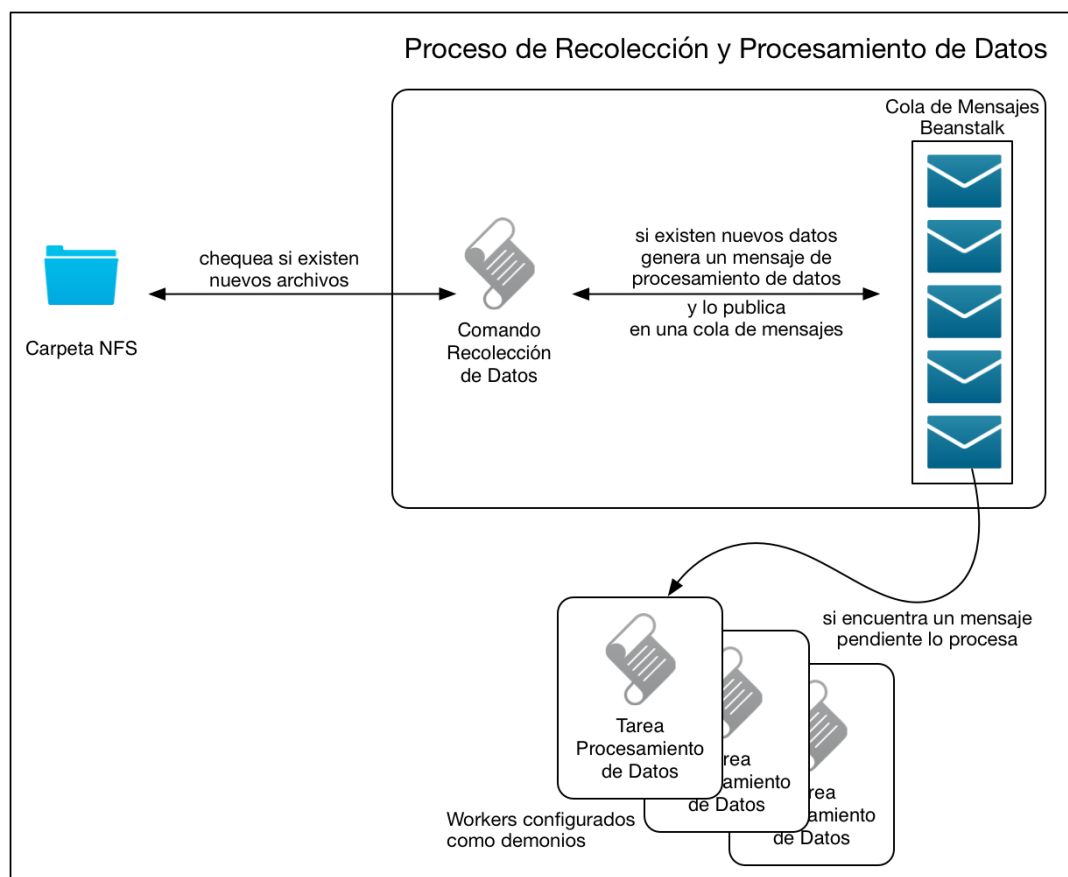


Ilustración 23: GEMHA - Proceso de Recolección y Procesamiento de Datos. Fuente: Elaboración propia.

Autenticación de usuarios

Para ingresar al Backend se requiere de una autenticación por medio de un nombre de usuario y una contraseña, esta aplicación permite la gestión de los usuarios dentro de la misma (ver Ilustración 24).

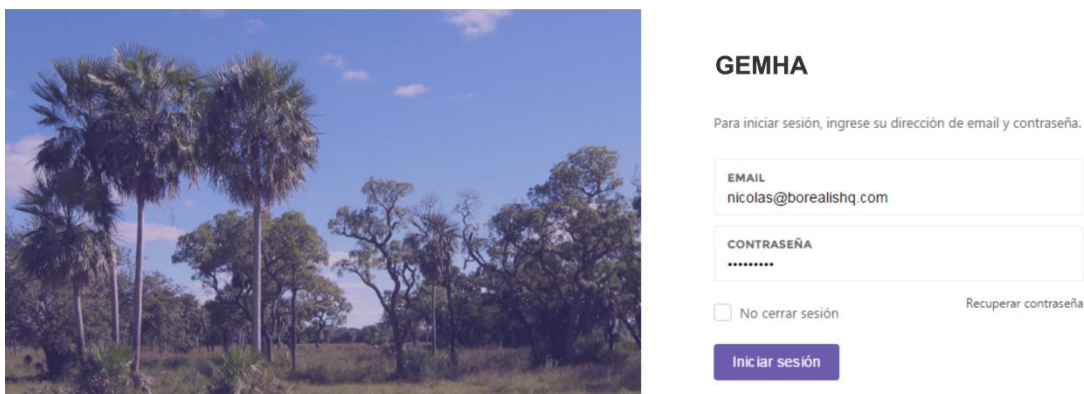


Ilustración 24: GEMHA - Ingreso al sistema. Fuente: Elaboración propia.

Tipos de estaciones

La aplicación permite gestionar los distintos tipos de estaciones y asignar el analizador de datos correspondiente para ese tipo de estación. En la Ilustración 25 se puede ver el listado de tipos de estaciones del sistema.

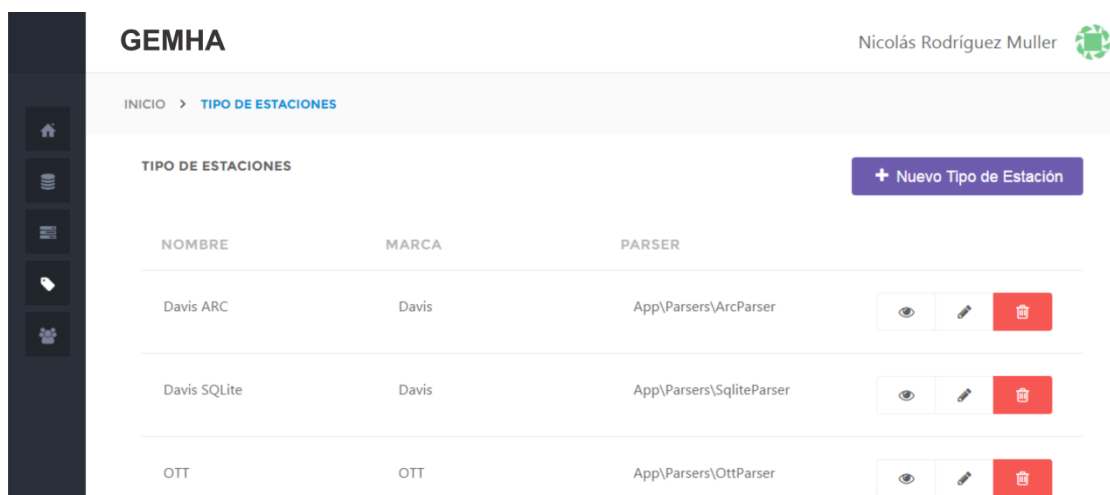


Ilustración 25: GEMHA - Tipos de estaciones. Fuente: Elaboración propia.

Estaciones

La aplicación permite gestionar los metadatos de las estaciones y sus estados. En la Ilustración 26 se puede apreciar la interfaz donde se listan las estaciones del sistema.

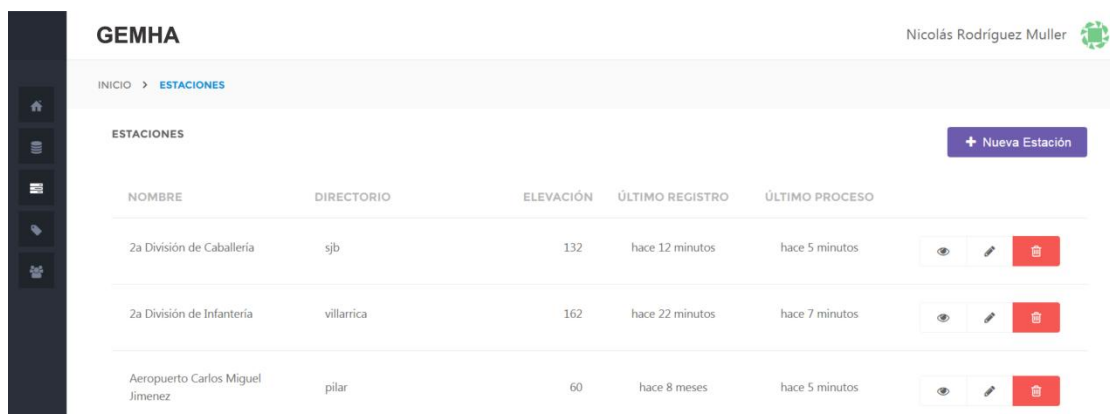
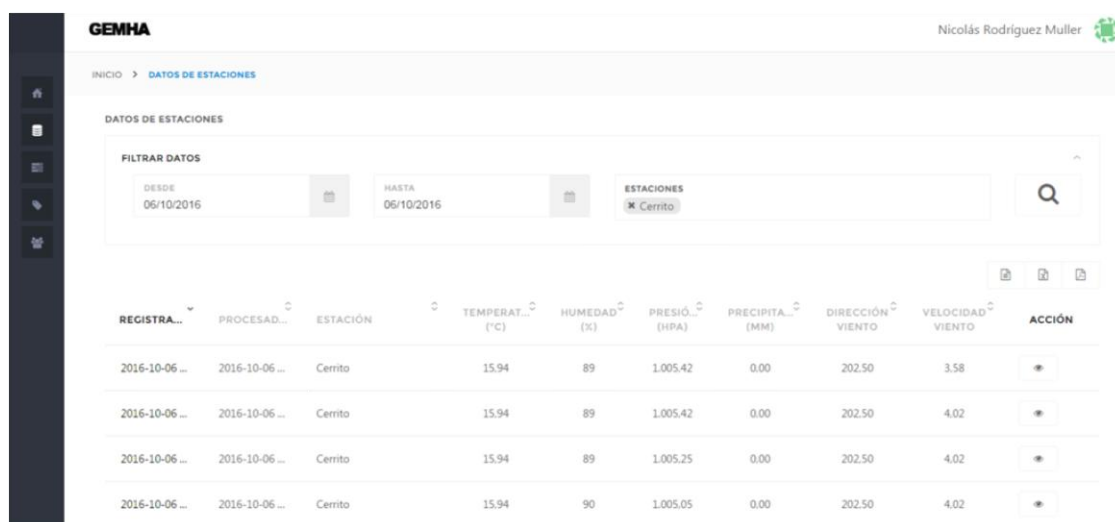


Ilustración 26: GEMHA - Gestión de Estaciones. Fuente: Elaboración propia.

Datos de estaciones

La aplicación permite visualizar y exportar en formato CSV los datos registrados por las estaciones. Además, provee opciones que permiten filtrar los datos por rango de fechas y estaciones (Ilustración 27). En esta sección se pueden visualizar rápidamente aquellos registros que fueron marcados como inválidos o erróneos cuyos datos no se ajustan a las reglas de validación.



The screenshot shows the GEMHA web application interface. The top header includes the GEMHA logo and the user name 'Nicolás Rodríguez Muller'. The main content area is titled 'DATOS DE ESTACIONES' and features a 'FILTRAR DATOS' section with filters for 'DESDE' (06/10/2016), 'HASTA' (06/10/2016), and 'ESTACIONES' (Cerrito). Below the filters is a table with the following columns: REGISTRA..., PROCESAD..., ESTACIÓN, TEMPERAT... (°C), HUMEDAD... (%), PRESIÓ... (HPA), PRECIPITA... (MM), DIRECCIÓN VIENTO, VELOCIDAD VIENTO, and ACCIÓN. The table displays four rows of data for the station 'Cerrito' on 2016-10-06.

REGISTRA...	PROCESAD...	ESTACIÓN	TEMPERAT... (°C)	HUMEDAD... (%)	PRESIÓ... (HPA)	PRECIPITA... (MM)	DIRECCIÓN VIENTO	VELOCIDAD VIENTO	ACCIÓN
2016-10-06 ...	2016-10-06 ...	Cerrito	15.94	89	1.005.42	0.00	202.50	3.58	*
2016-10-06 ...	2016-10-06 ...	Cerrito	15.94	89	1.005.42	0.00	202.50	4.02	*
2016-10-06 ...	2016-10-06 ...	Cerrito	15.94	89	1.005.25	0.00	202.50	4.02	*
2016-10-06 ...	2016-10-06 ...	Cerrito	15.94	90	1.005.05	0.00	202.50	4.02	*

Ilustración 27: GEMHA - Datos de Estaciones. Fuente: Elaboración propia.

9.1.3.2 Componente frontend o página web principal

Componente de acceso público que permite visualizar los datos actuales de las estaciones y adicionalmente, permitir a usuarios registrarse en la aplicación y solicitar acceso a la API. Cuenta con los siguientes sub componentes:

Página de inicio

En esta página web se visualizan las ubicaciones de distintas estaciones registradas en el sistema en un mapa, seleccionando cualquiera de ellas se pueden observar los últimos datos registrados, tal como se muestra en la Ilustración 28.

En el mapa se visualizan las distintas estaciones, estas se muestran de varios colores que representan el estado de cada una:

- **Verde:** la estación transmitió datos en los últimos 10 minutos.
- **Naranja:** se transmitió datos en las últimas 12 horas.
- **Rojo:** la última transmisión fue hace más de 12 horas o se encuentra inactiva.



Ilustración 28: GEMHA - Página de inicio. Fuente: Elaboración propia.

Página de estación

Permite la visualización de información detallada de la estación:

- Metadatos de la estación.
- Últimos datos observados.
- Históricos registrados (en formato tabular y de gráficos, de treinta días pasados) de temperatura, humedad, precipitación, radiación, presión y velocidad del viento para las estaciones meteorológicas. En las hidrológicas se muestra el nivel y la temperatura del agua.

Una muestra de la página de estación se puede apreciar en la Ilustración 29.

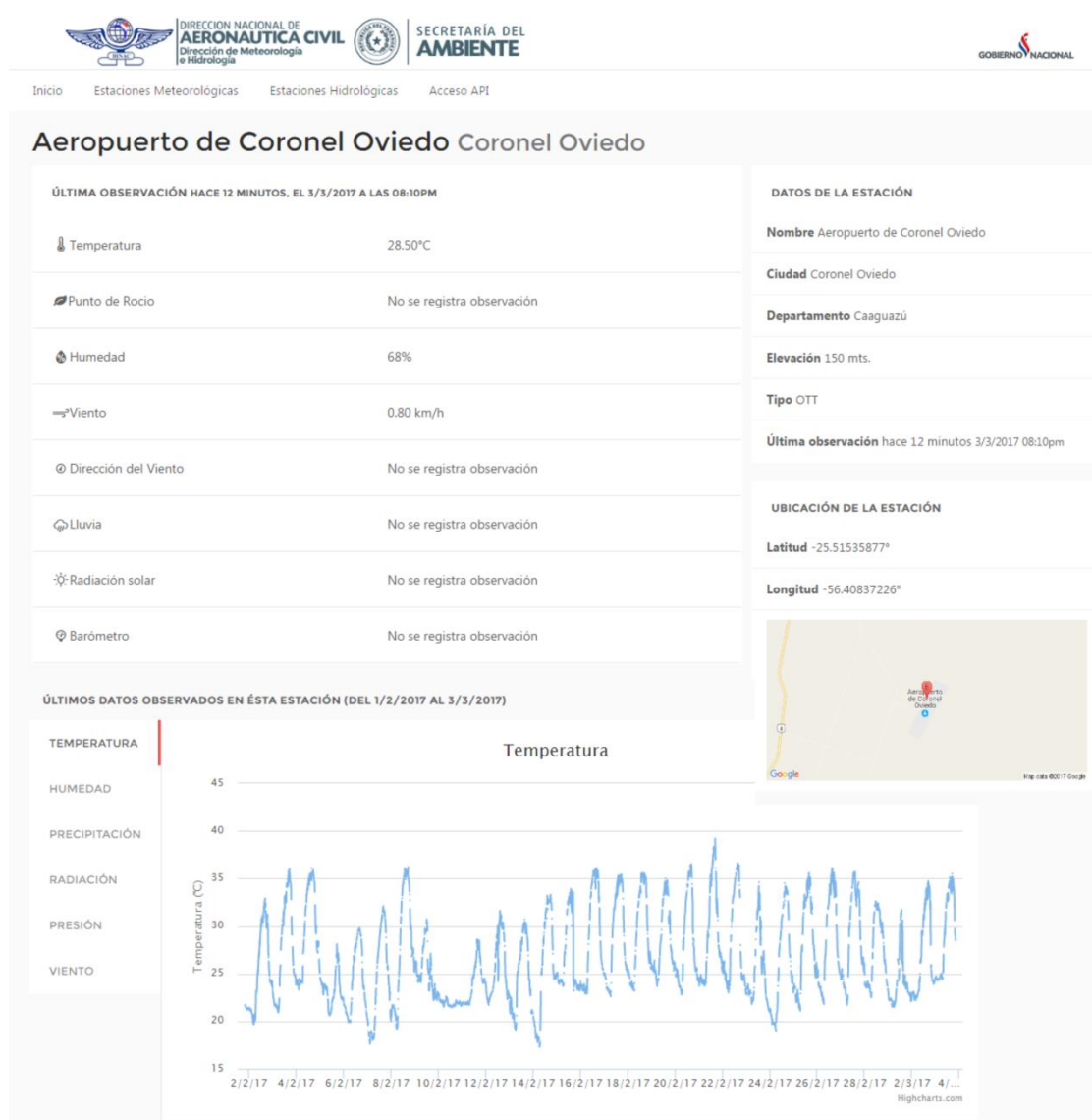


Ilustración 29: GEMHA - Página Estación. Fuente: Elaboración propia.

9.2 WRF Runner

El WRF Runner es un sistema desarrollado para configurar, automatizar y monitorear las simulaciones del WRF a intervalos regulares.

9.2.1 Proceso de ejecución del modelo WRF

Antes de realizar una simulación, se define el dominio principal y los sub-dominios (ver punto 3.6) que serán abarcados por el modelo.

El WRF tiene dos componentes principales que son el WPS (o WRF Pre-Processing System) y el WRF. Ambos obtienen los parámetros con los que deben operar por medio de archivos de configuración. [37]

- **WPS:** Componente que se encarga de preparar los archivos de entrada para el modelo.
- **WRF:** se encarga de hacer la simulación, utilizando los archivos de entrada generados por el WPS.

En este contexto, el flujo para correr una simulación para un rango de fechas definido se realiza de la siguiente forma:

- Se descargan las condiciones de borde para la cantidad de horas que se desea simular.
- Se definen los parámetros de la simulación (cantidad de dominios, fecha de inicio y fin, características de los dominios, etc.) en el archivo de configuración del WPS.
- Se ejecuta *geogrid.exe* para generar los datos de terreno estáticos.
- Se vinculan (como enlaces simbólicos) los archivos de las condiciones de borde a la carpeta del WPS.
- Se ejecuta *ungrib.exe* para desempaquetar los datos de las condiciones de borde y re-empaquetarlos en un formato intermedio.
- Se ejecuta *metgrid.exe* para interpolar los datos horizontalmente en el dominio definido.
- Se enlazan (como enlaces simbólicos) los archivos generados por *metgrid.exe* a la carpeta del WRF.
- Se definen los parámetros de la simulación (cantidad de dominios, fecha de inicio y fin, características de los dominios, parametrizaciones físicas, etc.) en el archivo de configuración del WRF.
- Se ejecuta *real.exe* para interpolar los datos verticalmente a las coordenadas del modelo.
- Se ejecuta *wrf.exe* para correr la simulación y generar el pronóstico.

Una vez finalizada la ejecución del modelo, se crea un archivo netCDF con el pronóstico del tiempo para el rango de fechas seleccionado en el dominio configurado.

Posteriormente, se debe pos procesar este archivo a fin de extraer los resultados para los puntos geográficos específicos e insertarlos en una base de datos para su posterior consulta.

9.2.2 Componentes del sistema

El sistema WRF Runner automatiza el flujo de ejecución del modelo, desde la descarga de los archivos del modelo GFS, que son utilizados como condición de borde, hasta el pos procesamiento de la salida del modelo.

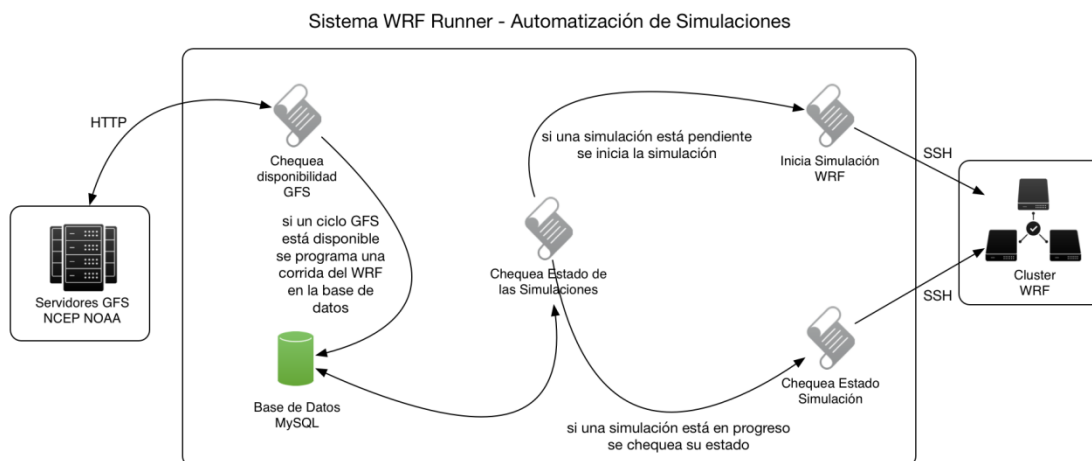


Ilustración 30: WRF Runner – Automatización de Simulaciones. Fuente: Elaboración propia.

9.2.2.1 GFS Downloader

Este sub componente se encarga de verificar el estado de ejecución de cada ciclo del Modelo GFS (ver punto 4.4). Cuando un ciclo ha terminado, comprueba la disponibilidad de los archivos generados; si encuentra disponibles, inicia y monitorea la descarga de los mismos. Una vez finalizada toda la descarga, programa la ejecución de la simulación con los nuevos datos obtenidos.

9.2.2.2 Monitor de simulaciones WRF

Se encarga de controlar si existen simulaciones programadas pendientes; en su caso, prepara los archivos de configuración para la simulación y ejecuta la corrida del modelo siguiendo los siguientes pasos:

- Limpia las carpetas del WPS y del WRF, borrando cualquier archivo remanente de una simulación anterior.
- Crea las carpetas donde se almacenarán las configuraciones y datos de la simulación.
- Prepara y crea los archivos de configuración para el WPS y el WRF.
- Verifica la existencia de la cantidad necesaria de archivos de condición de borde para el número de horas de simulación requeridas.
- Crea los enlaces simbólicos a los archivos de configuración en las carpetas WPS y WRF respectivamente.
- Ejecuta el programa ungrib.exe.

Adicionalmente, si una simulación se encuentra en progreso, es responsable de lanzar el sub componente de estado de la simulación.

9.2.2.3 Estado de la simulación

Da seguimiento al estado de una simulación en progreso. Es el sub componente principal de la aplicación, ya que es el encargado de verificar el progreso del flujo de ejecución del modelo (ver punto 4.5) y a su vez, de capturar datos estadísticos como hora de inicio y fin del proceso, cálculo de tiempo de ejecución y captura de la salida.

Como el modelo no cuenta con una API o interfaz de consulta de estado, este sub componente realiza la tarea conectándose vía SSH al clúster a intervalos regulares de tiempo y, dependiendo del proceso que se encuentre en ejecución, chequea el archivo de registro (log) para comprobar si el mismo sigue ejecutándose. Una vez finalizado el proceso, toma la fecha/hora de modificación del log (que es el momento en que finalizó la tarea) y calcula el tiempo de ejecución del mismo. Seguidamente, ejecuta el siguiente proceso y vuelve a monitorearlo.

Los procesos monitoreados por este sub componente son el *ungrib.exe*, *metgrid.exe*, *real.exe* y *wrf.exe*, cada uno de ellos genera logs de distintos formatos.

Una vez finalizada la ejecución del modelo, se encarga de ejecutar el pos procesamiento de los datos. El mismo se encarga de extraer las variables meteorológicas del pronóstico generado para los sitios definidos en el sistema. Estas variables son almacenadas en una base de datos MySQL.

9.2.2.4 Interfaz de administración

El sistema es administrable a través de una interfaz que permite al usuario ingresar con sus credenciales de acceso. Puede tener acceso a diversas funciones:

Escritorio: muestra un gráfico estadístico con datos que incluyen la cantidad de simulaciones por día y el tiempo promedio de cada proceso, tal como se puede apreciar en la Ilustración 31.

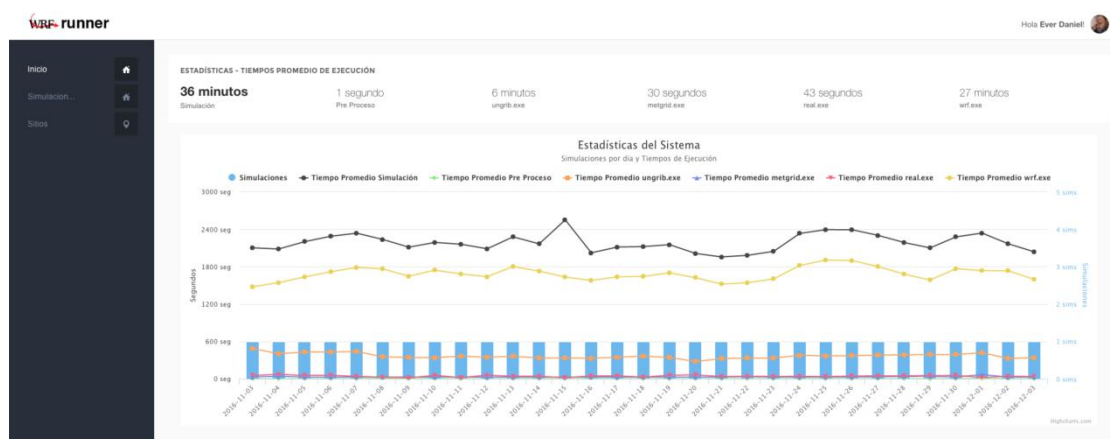


Ilustración 31: WRF Runner - Dashboard. Fuente: Elaboración propia.

Previsión meteorológica del tiempo para Encarnación en un clúster de Raspberry Pi's utilizando el Modelo de Investigación y Pronóstico del Tiempo (WRF)

Simulaciones: interfaz que permite ver las diversas ejecuciones del modelo realizadas, el estado de las mismas, estadísticas y una opción que muestra los detalles específicos de cada una, como ser: fecha/hora de inicio y fin de cada proceso, duración de cada uno, y la visualización de los archivos de registros generados (ver Ilustración 32 e Ilustración 33).

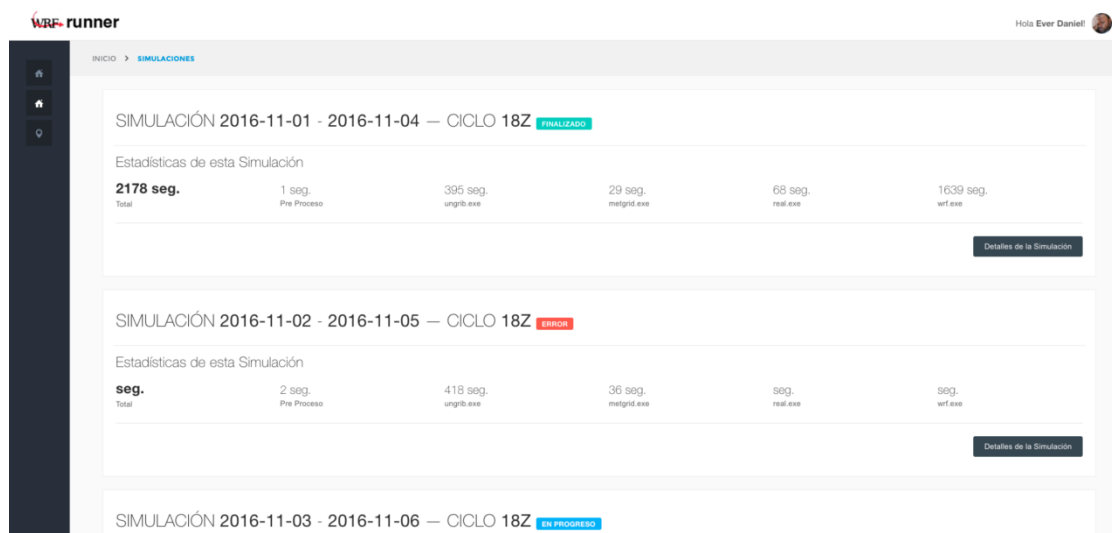


Ilustración 32: WRF Runner - Simulaciones. Fuente: Elaboración propia.

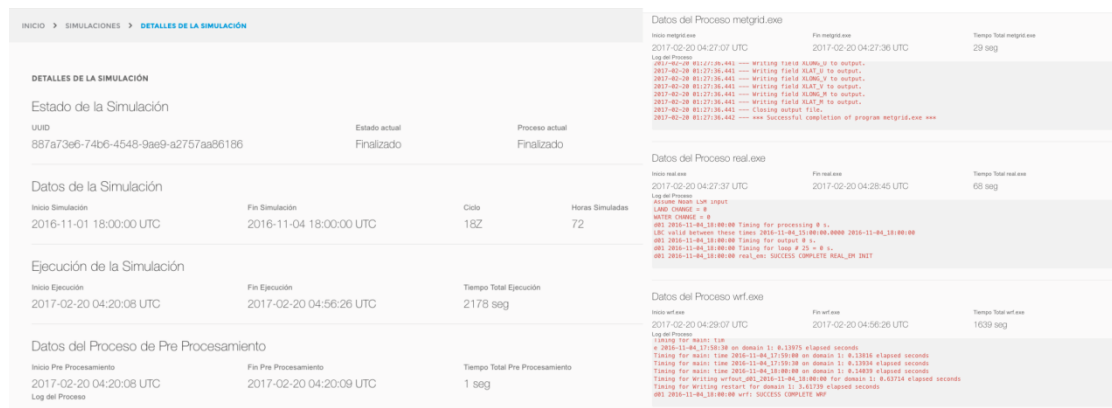
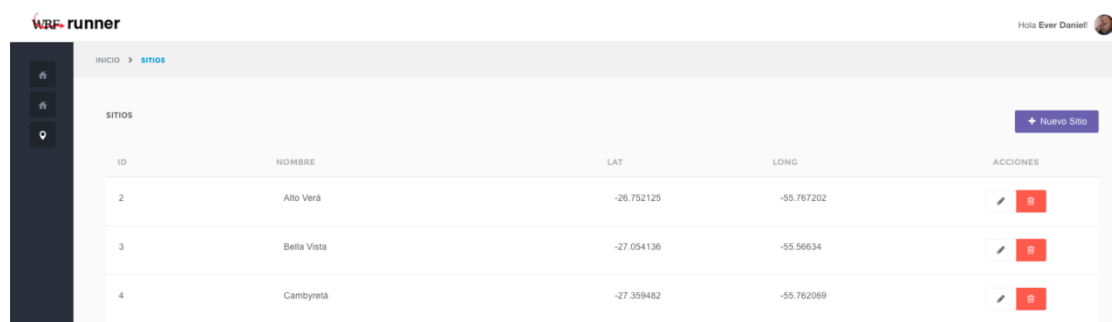


Ilustración 33: WRF Runner - Detalles de Simulación. Fuente: Elaboración propia.

Gestión de sitios: en esta sección se pueden administrar los sitios cuyas coordenadas geográficas serán utilizadas a la hora de correr el pos procesamiento de datos para obtener las variables de pronóstico para cada ubicación. Estos sitios se muestran en la Aplicación Móvil.



The screenshot shows the 'WRF-runner' web application interface. At the top, there's a navigation bar with 'INICIO' and 'SITIOS'. Below this, a table lists three sites. Each site entry includes an ID, a name, latitude (LAT), longitude (LONG), and a set of actions (edit and delete icons). A '+ Nuevo Sitio' button is located in the top right corner of the table area.

ID	NOMBRE	LAT	LONG	ACCIONES
2	Alto Verá	-26.752125	-55.767202	[edit] [delete]
3	Bella Vista	-27.054136	-55.56634	[edit] [delete]
4	Cambyretá	-27.359482	-55.762069	[edit] [delete]

Ilustración 34: WRF Runner - Sitios. Fuente: Elaboración propia.

9.2.2.5 API de acceso a datos pronosticados

Esta API permite acceder programáticamente a los pronósticos generados por el modelo WRF para cada sitio geográfico definido en el sistema, por cada fecha y hora de simulación. Las variables generadas para cada punto geográfico son:

- Temperatura a 2 metros del suelo.
- Presión atmosférica a nivel del mar.
- Velocidad del viento a 10 metros.
- Dirección del viento a 10 metros.
- Precipitación acumulada en la última hora.

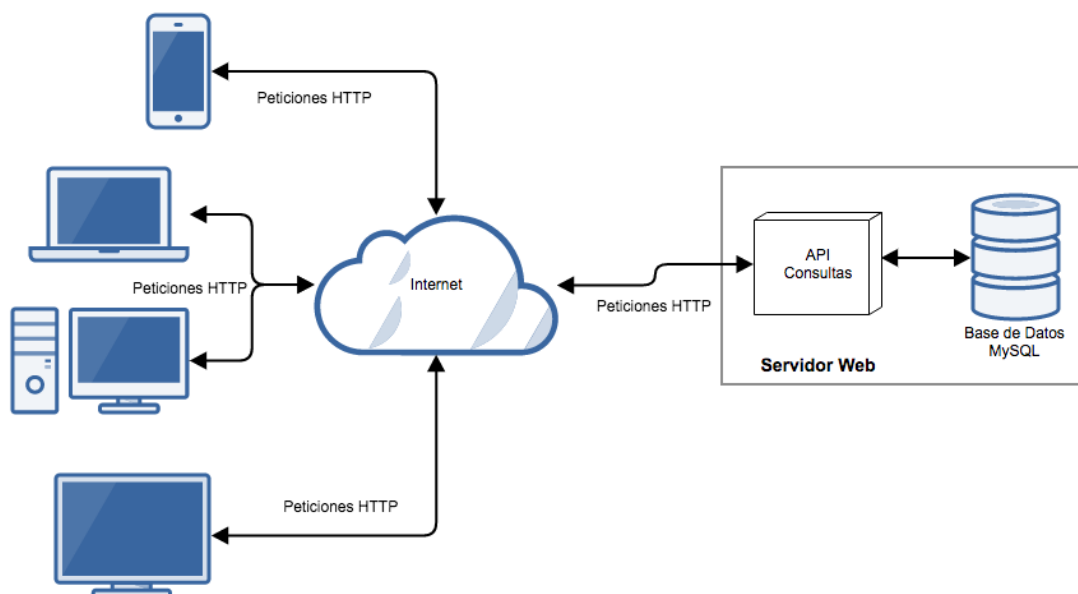


Ilustración 35: Esquema conceptual del Servidor Web y la API. Fuente: Elaboración propia.

9.3 Aplicación Móvil

Como se mencionó en el punto 9.2, el WRF Runner es el encargado de automatizar la ejecución del modelo, de pos procesar y de almacenar en una base de datos los pronósticos obtenidos.

Con la aplicación móvil se cierra el ciclo, permitiendo que el usuario final pueda visualizar de forma rápida y fácil el pronóstico del tiempo para cada uno de los distritos del Departamento de Itapúa.

9.3.1 Análisis

La visualización del pronóstico del tiempo es de acceso público; por tanto, la aplicación móvil no requerirá de ningún inicio de sesión ni que el usuario registre sus datos para utilizar la aplicación.

Sin embargo, se permitirá que el usuario pueda personalizar la aplicación para adaptarlo a sus necesidades.

9.3.2 Detalles del sistema

La aplicación móvil desarrollada permite acceder de forma rápida al pronóstico del tiempo, a 3 días, para los 30 Distritos que conforman el Departamento de Itapúa (ver Ilustración 36).



Ilustración 36: APP - Pronóstico de distrito.
Fuente: Elaboración propia.

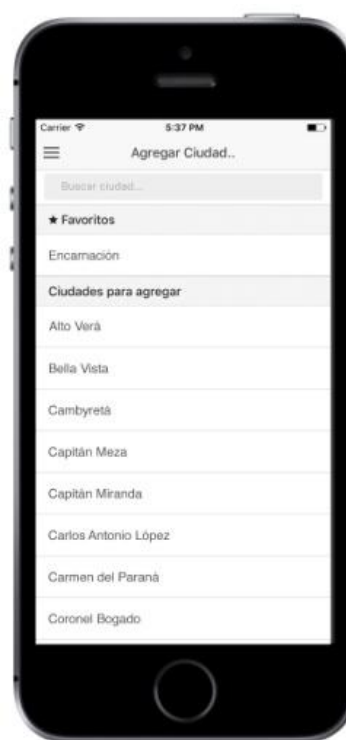


Ilustración 37: APP - Selección de distritos.
Fuente: Elaboración propia.

Por cada Distrito, la aplicación muestra los siguientes datos:

Para la fecha y hora actual

- Temperatura pronosticada.
- Velocidad del viento pronosticada.
- Dirección del viento pronosticada.
- Cantidad de Precipitación pronosticada.

Para los siguientes 3 días

- Temperatura máxima y mínima del día.
- Velocidad máxima del viento.
- Dirección del viento predominante.
- Precipitación acumulada del día.

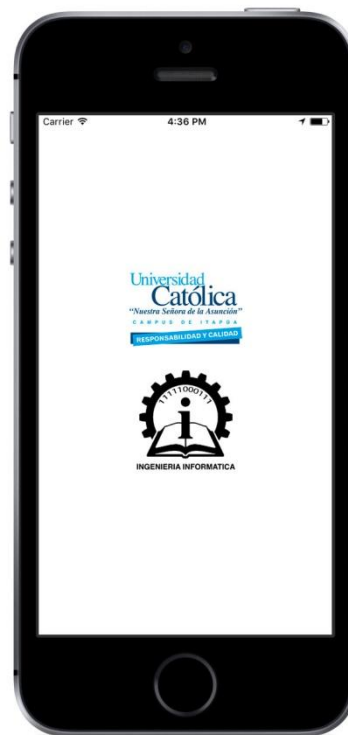


Ilustración 38: APP - Pantalla inicial.
Fuente: Elaboración propia.

El usuario puede además personalizar la aplicación, pudiendo definir los distritos de su interés, con esto, la aplicación únicamente mostrará los datos para los distritos seleccionados (ver Ilustración 37).

10 Pruebas realizadas

10.1 GEMHA

Como se menciona en el punto 9.1.1, se realizó un inventario de las estaciones meteorológicas e hidrológicas automáticas que componen la red de estaciones de la DMH (ver punto 12.1).

Las estaciones automáticas más antiguas han empezado a generar datos desde el año 2014, a intervalos de 10 minutos, lo que totaliza alrededor de 7GB de datos a procesar (ver punto 12.1.3), con lo que surgen dos desafíos; procesar, por una parte, en tiempo real los datos que envían las estaciones, y por la otra, los datos históricos existentes.

La DMH disponibilizó un servidor virtual con 2 núcleos para el despliegue del sistema GEMHA. Como paso previo a la publicación del sistema, se realizó una serie de pruebas para verificar que los recursos proveídos sean suficientes para atender la demanda de procesamiento de los datos transmitidos por la Red de Estaciones.

10.1.1 Proceso de datos de estaciones

10.1.1.1 Prueba 1

Descripción

Prueba del procesamiento de datos en el servidor. Se configuró 2 tareas que se ejecutan como demonios (daemons) en el servidor, empleando la aplicación *supervisord*.

Resultados

Se observó que los recursos proveídos no fueron suficientes para procesar los datos de forma rápida, ya que las órdenes de proceso enviadas a las tareas demoraban alrededor de 17 minutos en ser procesadas. Se procedió a solicitar más recursos de procesamiento a la DMH.

10.1.1.2 Prueba 2

Descripción

Tras la solicitud realizada luego de la Prueba 1, la DMH incrementó los recursos de procesamiento del servidor virtual, elevando el número de núcleos de 2 a 8. Se realizó otra prueba de proceso de los datos.

Resultados

El proceso de datos se ejecutó de forma satisfactoria. El incremento en los recursos de procesamiento del servidor virtual permitió procesar los datos de forma rápida, atendiendo las órdenes de proceso de manera oportuna. Como se puede observar en las métricas de Job Age, que miden el tiempo que tarda en atenderse una orden, y Queue Size que miden el tamaño de la cola, en donde muchos mensajes indican un retardo en el proceso de los mismos (ver Ilustración 39).

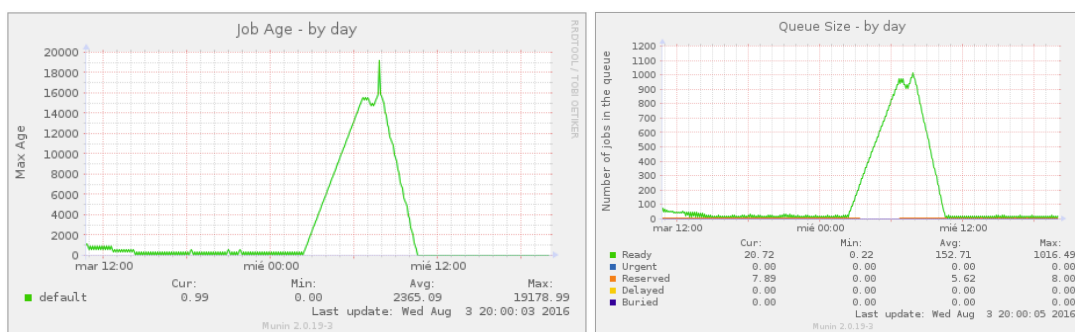


Ilustración 39: GEMHA - Job Age y Queue Size. Fuente: Elaboración propia.

10.2 Compilación del WRF en el Raspberry Pi

Se efectuaron pruebas graduales para compilar el WRF en la arquitectura ARM.

10.2.1 Compilación y prueba del MPI

10.2.1.1 Prueba 1

Descripción

Compilar el MPI para la arquitectura ARM. Para el efecto, se siguió un tutorial publicado por la Facultad de Ingeniería y Medioambiente de la Universidad de Southampton [38].

Para realizar esta prueba, se emplearon cuatro RPi's, designando uno de ellos como nodo de acceso (LN) y los demás como nodos de cómputo.

Resultados

Al final de la prueba se consiguió compilar correctamente la implementación del MPI en la arquitectura ARM, y correr una aplicación de ejemplo que ejecuta un proceso en todos los nodos.

Como resultado adicional se pudo dimensionar la complejidad que conllevaría construir el clúster.



Ilustración 40: Primer Clúster RPi. Fuente: Elaboración propia.

10.2.2 Compilación del WRF

10.2.2.1 Prueba 1

Descripción

Confirmado que el MPI funciona correctamente en la arquitectura ARM, se dio lugar a las pruebas de compilación del WRF en el Raspberry Pi.

Se inició con el WRF 3.8.1, que es la versión más reciente lanzada en el mes de agosto de 2016.

El proceso de compilación, tanto del WRF como del WPS, está compuesto de dos pasos; el primero trata de configurar el modelo para su compilación, donde se elige el

compilador a ser utilizado y si se desea habilitar soporte MPI o no; y el segundo, es la compilación en sí del modelo.

Resultados

Se pudo notar que el WRF 3.8.1 no posee parámetros de compilación determinados por defecto para la arquitectura ARM, por lo que se emplearon los establecidos para la arquitectura “Linux i486 i586 i686”. Esta arquitectura utiliza parámetros para *gfortran* y *gcc*; a raíz de esto, se consideró que pueden ser compatibles, pues los mismos se encuentran disponibles en el Sistema Operativo Raspbian.

Una vez definidos los parámetros de compilación para la arquitectura ARM se pudo finalmente configurar el WRF para su compilación.

10.2.2.2 Prueba 2

Descripción

Se buscó compilar el modelo utilizando los parámetros de configuración definidos en la prueba anterior.

Resultados

Esta prueba dio un resultado negativo, debido a que no se pudo compilar el WRF 3.8.1 en el Raspberry Pi, porque el proceso de compilación arrojó varios errores. Entonces se probó compilar las demás versiones del WRF, empezando por el WRF 3.8 hasta la versión 3.2.1, obteniendo nuevamente los mismos resultados.

10.2.2.3 Prueba 3

Descripción

A raíz de los resultados arrojados por la prueba anterior, se cambiaron los parámetros de compilación en busca de una combinación que logre realizar la tarea sin errores.

Resultados

En esta prueba tampoco se pudo compilar el WRF. El proceso terminaba abruptamente y al inspeccionar el log de compilación se notaban varios errores del compilador acerca de archivos de entrada que no estaban completos. Al inspeccionar los archivos utilizados por el compilador, se notó que varios de ellos no estaban completos, y en algunos casos hasta se encontraban vacíos.

10.2.2.4 Prueba 4

Descripción

En atención al resultado anterior, se habilitó una característica del compilador que permitiría la utilización de todos los núcleos disponibles en el procesador para realizar la compilación.

Resultados

No se consiguió compilar el WRF. Se incrementó el número por defecto de núcleos a utilizar por el compilador de 2 a 4 (el total de núcleos disponibles en el RPi) y se visualizaron los mismos errores que en la prueba anterior.

10.2.2.5 Prueba 6

Descripción

Luego de una investigación exhaustiva de los errores de compilación arrojados en las pruebas anteriores, se dedujo que posiblemente los archivos compilados no estaban completos o en algunos casos vacíos, porque el RPi no contaba con suficiente memoria. En esta prueba, se pretendió determinar el problema e intentar solucionarlo.

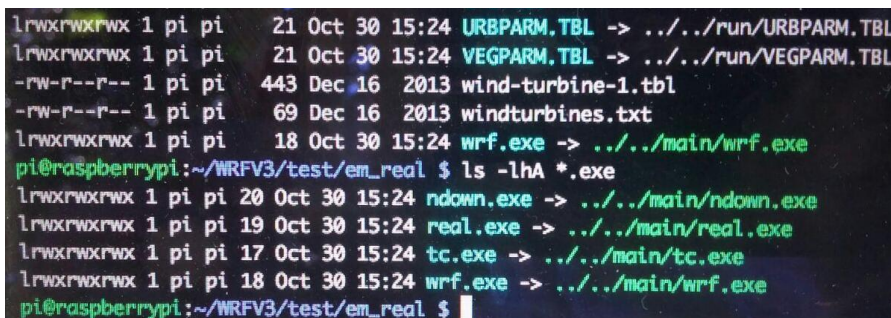
Resultados

Al compilar el WRF, se esperó que genere los errores de las pruebas anteriores. Una vez que la compilación falló, se inspeccionó el volcado de mensajes del núcleo del procesador utilizando la herramienta *dmesg*, donde finalmente se confirmó que el proceso de compilación fue matado (process killed) por el sistema operativo por un error de “Sin Memoria” (Out of Memory).

Entonces se incrementó el tamaño del SWAP (Área de Intercambio) del sistema operativo, de los 100MB definidos por defecto a 1024MB (1GB).

Con este ajuste, el proceso de compilación finalizó correctamente y se generaron los archivos binarios ejecutables del modelo para la arquitectura ARM.

Posteriormente, se restauró la configuración del SWAP a su valor anterior de 100MB.



```
lrwxrwxrwx 1 pi pi 21 Oct 30 15:24 URBPARM.TBL -> ../../run/URBPARM.TBL
lrwxrwxrwx 1 pi pi 21 Oct 30 15:24 VEGPARM.TBL -> ../../run/VEGPARM.TBL
-rw-r--r-- 1 pi pi 443 Dec 16 2013 wind-turbine-1.tbl
-rw-r--r-- 1 pi pi 69 Dec 16 2013 windturbines.txt
lrwxrwxrwx 1 pi pi 18 Oct 30 15:24 wrf.exe -> ../../main/wrf.exe
pi@raspberrypi:~/WRFV3/test/em_real $ ls -lha *.exe
lrwxrwxrwx 1 pi pi 20 Oct 30 15:24 ndown.exe -> ../../main/ndown.exe
lrwxrwxrwx 1 pi pi 19 Oct 30 15:24 real.exe -> ../../main/real.exe
lrwxrwxrwx 1 pi pi 17 Oct 30 15:24 tc.exe -> ../../main/tc.exe
lrwxrwxrwx 1 pi pi 18 Oct 30 15:24 wrf.exe -> ../../main/wrf.exe
pi@raspberrypi:~/WRFV3/test/em_real $
```

Ilustración 41: Compilación WRF en Arquitectura ARM. Fuente: Elaboración Propia.

10.3 Configuración del clúster para el WRF

A continuación se configuró la carpeta del WRF, para compartir y montar la misma en los nodos de cómputo.

10.3.1 Configuración del NFS

10.3.1.1 Prueba 1

Descripción

En esta prueba se buscó configurar una carpeta como compartida a través del NFS en el nodo de acceso, a fin de que los nodos de cómputo accedan a los archivos binarios del modelo WRF.

Resultados

Se compartió correctamente la carpeta */wrf*, definiéndola como compartida en el archivo *exports* del nodo de acceso y reiniciando el servidor NFS (*nfs-kernel-server*).

10.3.2 Configuración de los clientes NFS

10.3.2.1 Prueba 1

Descripción

Con la carpeta *wrf* compartida en el nodo de acceso, se intentó montar la misma en cada uno de los nodos de cómputo del clúster.

Resultados

No se logró montar la carpeta compartida en los nodos del clúster. El comando de montar la carpeta devuelve un error después de unos segundos.

10.3.2.2 Prueba 2

Descripción

Como el resultado de la prueba anterior fue negativo, es decir, no se logró montar la carpeta compartida en el nodo de acceso en los nodos de cómputo. Se vuelve a realizar otra, después de investigar los errores devueltos, esta vez reiniciando los servidores NFS (*nfs-kernel-server*) y RPC Bind (*rpcbind*) antes de realizar la prueba.

Resultados

Se logró montar correctamente la carpeta en los nodos de cómputo. La verificación es efectuada en todos los nodos ejecutando el comando *mount* e inspeccionando la salida del comando.

10.3.3 Verificación de ejecución del modelo

10.3.3.1 Prueba 1

Descripción

Con las carpetas compartidas y disponibles en los nodos de cómputo, se continuó con la ejecución de prueba del modelo en el clúster, utilizando los 20 nodos con todos sus núcleos, totalizando 80 núcleos.

Resultados

La ejecución del modelo en el clúster falló después de unos minutos. Se verificó que el modelo no está en ejecución al mirar los procesos activos en el nodo de acceso con el comando *ps*. Se procedió a inspeccionar los logs del modelo, no encontrándose error de ejecución aparente.

10.3.3.2 Prueba 2

Descripción

En base a la prueba anterior, se probó ejecutar el modelo con un número distinto de núcleos asignados a la ejecución.

Resultados

Se vuelve a constatar que la ejecución del modelo falla después de unos minutos. Se observó que se trataba de un problema similar al de la prueba anterior, pues no se registran errores de ejecución en los logs.

10.3.3.3 Prueba 3

Descripción

Con los antecedentes de las pruebas anteriores, se investigó si consistía en un problema de memoria, por lo que se vuelve a correr el modelo para inspeccionar el volcado de mensajes del proceso en uno de los nodos de cómputo.

Resultados

Luego de verificar varios nodos, en uno de ellos se verificó que efectivamente se trataba de un problema de memoria, se visualizaron los mensajes del sistema operativo matando (process killed) el binario en ejecución del modelo en el nodo por falta de memoria (Out of Memory). Se puede ver el error en la Ilustración 42.

Se decide incrementar el SWAP del sistema operativo a la misma cantidad de RAM disponible en el sistema, por lo que se cambia el valor de 100MB a 1024MB

```
[ 4074.048444] [ 797]  0  797  1303  71  6  0  26  0 htop
[ 4074.048463] [ 838]  0  838  2838  0  9  0  151  0 sshd
[ 4074.048479] [ 843]  0  843  877  1  6  0  162  0 hydra_pmi_proxy
[ 4074.048495] [ 845]  0  845  122183  61328  169  0  4577  0 wrf.exe
[ 4074.048511] [ 846]  0  846  122307  35842  125  0  6006  0 wrf.exe
[ 4074.048526] [ 847]  0  847  122494  68790  188  0  6011  0 wrf.exe
[ 4074.048541] [ 848]  0  848  121872  67166  185  0  5657  0 wrf.exe
[ 4074.048554] Out of memory: Kill process 847 (wrf.exe) score 264 or sacrifice child
[ 4074.048577] Killed process 847 (wrf.exe) total-vm:489976kB, anon-rss:274280kB, file-rss:880kB
[ 4525.163159] Adding 1048572k swap on /var/swap. Priority:-1 extents:7 across:1458172k SSFS
```

Ilustración 42: Proceso matado por falta de memoria. Fuente: Elaboración propia.

10.3.3.4 Prueba 4

Descripción

Incrementado el SWAP en el sistema operativo a la misma cantidad de RAM disponible en el sistema, se vuelve a correr el modelo en el clúster.

Resultados

El modelo finalmente se ejecuta de forma satisfactoria en el clúster. Se analizó la salida del modelo, inspeccionando que los días de simulación configurados estén disponibles en el archivo de salida utilizando el comando *netCDF* “*ncdump*”.

10.3.4 Tiempos de ejecución

10.3.4.1 Prueba 1

Descripción

Con el modelo ejecutándose correctamente en el clúster, se procedió a realizar pruebas de los tiempos de ejecución utilizando un periodo de simulación de 6 horas y cantidades distintas de núcleos (64, 72 y 80 núcleos).

Resultados

Se comprobó que los tiempos de ejecución varían de acuerdo a la cantidad de núcleos asignados a la ejecución. Los tiempos de ejecución tienen una relación directa con, la cantidad de grillas definidas dentro del dominio y la cantidad de núcleos asignados [37].

Como se observa en la Tabla 5, la diferencia del tiempo de ejecución entre 64 y 80 núcleos es mínima, pero puesto que se cuentan con los recursos, se decidió utilizar todos los nodos a la hora de correr las simulaciones.

Cantidad Núcleos	Hora Inicio	Hora Fin	Tiempo Ejecución
64	18:05	18:58	00:53
72	19:31	20:33	01:02
80	14:01	14:52	00:51

Tabla 5: Tiempos de ejecución del Modelo WRF en el Clúster de RPi's para un periodo de 6 horas.

10.4 Parametrización del WRF

Se realizaron pruebas de parametrización del modelo para el dominio configurado. Utilizando los parámetros por defecto disponibles en el WRF.

10.4.1 Parametrización del WRF

10.4.1.1 Prueba 1

Descripción

En esta prueba se buscó determinar si la parametrización por defecto disponible en el WRF es correcta para correr el modelo para el dominio configurado.

La parametrización por defecto del WRF 3.8.1 se detalla en la Tabla 6.

Parámetro	Valor
time_step	180
e_vert	30
p_top_requested	5000
num_metgrid_levels	27
num_metgrid_soil_levels	4
mp_physics	3
ra_lw_physics	1
ra_sw_physics	1
radt	30
sf_sfclay_physics	1
sf_surface_physics	2
bl_pbl_physics	1
cu_physics	5
num_soil_layers	4
sf_urban_physics	0

Tabla 6: Parametrización por defecto del WRF 3.8.1

Resultados

No se ejecutó completamente el modelo. Se generaron las salidas del WPS y al ejecutar real.exe generó el siguiente error:

```
----- FATAL CALLED -----  
FATAL CALLED FROM FILE: input_wrf.b LINE: 413  
input_wrf.F: SIZE MISMATCH: namelist ide,jde,num_metgrid_levels
```

Se inspeccionaron los archivos de salida generados por el proceso METGRID.EXE y se observó que el número de niveles de suelo generados en la salida del WPS eran 32, y no coincidía con el parámetro definido por defecto que era de 27.

10.4.1.2 Prueba 2

Descripción

En esta prueba se ajustó únicamente el parámetro *num_metgrid_levels* y se volvió a correr el modelo. La nueva parametrización se describe en la Tabla 7.

Parámetro	Valor Anterior	Nuevo Valor
num_metgrid_levels	27	32

Tabla 7: Parametrización del WRF - Prueba 2

Resultados

El proceso REAL.EXE volvió a fallar, generando el siguiente error en el archivo de registro:

```
FATAL CALLED FROM FILE: module_initialize_real.b LINE: 575  
not enough info for a p sfc computation
```

10.4.1.3 Prueba 3

Descripción

Se investigó el error generado en la prueba anterior y se encontró que es necesario incrementar el valor del parámetro *e_vert*. Se procedió a ajustar el valor como se describe en la Tabla 8 y se volvió a correr el modelo.

Parámetro	Valor Anterior	Nuevo Valor
e_vert	30	40

Tabla 8: Parametrización del WRF - Prueba 3

Resultados

En esta prueba se ejecutó el proceso REAL.EXE correctamente, verificado por el mensaje del proceso en el archivo de registro generado:

```
real_em: SUCCESS COMPLETE REAL_EM INIT
```

10.4.1.4 Prueba 4

Descripción

El proceso REAL.EXE en la prueba anterior generó los archivos de entrada necesarios para el WRF.EXE, por lo que se ejecutó el modelo.

Resultados

La ejecución del modelo no terminó de forma satisfactoria, generando avisos en el archivo de registro del WRF. Al investigar sobre estos avisos se determinó que los

errores se debían a un tipo conocido como *CFL Condition*. Luego de examinar exhaustivamente los demás registros se encontraron otros mensajes del tipo *Segmentation Fault*.

10.4.1.5 Prueba 5

Descripción

En la prueba anterior se pudo identificar los errores generados por lo que se procedió a hacer una investigación sobre los mismos y adicionalmente, revisar la documentación del Modelo. Finalmente se determinó que el parámetro *time_step* tenía un valor muy elevado para la configuración del dominio. La documentación del modelo recomienda que este parámetro sea definido como:

$$time_step = 6 * dx$$

Donde *dx* es el valor del tamaño horizontal de la grilla en kilómetros. [39]

Se procedió a definir un nuevo valor para el parámetro *time_step* en base al tamaño horizontal de la grilla del dominio, que es de 5 km. El nuevo valor se describe en la Tabla 9.

Parámetro	Valor Anterior	Nuevo Valor
time_step	180	30

Tabla 9: Parametrización del WRF - Prueba 5

Resultados

Se volvió a ejecutar el modelo, esta vez finalizó de forma satisfactoria.

10.4.1.6 Prueba 6

Descripción

En esta prueba se investigaron los parámetros físicos óptimos para un dominio de resolución de 5 km.

En la documentación del modelo se especifican los parámetros físicos sugeridos para grillas de 1 a 4 km [40], por lo que se procedió a configurar el dominio con los ajustes sugeridos.

Parámetro	Valor Anterior	Nuevo Valor
mp_physics	3	8
ra_lw_physics	1	4
ra_sw_physics	1	4
radt	30	10

Parámetro	Valor Anterior	Nuevo Valor
sf_sfclay_physics	1	2
sf_surface_physics	2	2
bl_pbl_physics	1	2
cu_physics	5	0

Tabla 10: Parámetros físicos WRF

Resultados

Se volvió a ejecutar el modelo, finalizando de forma satisfactoria con los parámetros ajustados.

10.4.1.7 Prueba 7

Descripción

Con los parámetros ajustados y definidos para el dominio, se realizó una ejecución del modelo en el Clúster, utilizando los datos del GFS del día 3 de noviembre de 2016, para un periodo de simulación de 72 horas.

Resultados

Se pudo ejecutar satisfactoriamente el modelo para el dominio configurado en el clúster. En la Tabla 11 se puede apreciar el tiempo de ejecución.

Cantidad Núcleos	Hora Inicio	Hora Fin	Tiempo Ejecución
80	17:35	02:27	08:52

Tabla 11: Tiempo de ejecución del Modelo WRF en el Clúster de RPi's para un periodo de 72 horas

10.5 WRF Runner

Las pruebas realizadas al WRF Runner requirieron dos servidores a fin de verificar si el sistema puede, conectarse vía SSH al servidor remoto y ejecutar comandos en el mismo.

Para el efecto, se utilizó el servidor virtual del sistema GEMHA y el servidor de pruebas, ambos proveídos por la DMH. En el servidor virtual se hizo el despliegue del WRF Runner, mientras que en el servidor de pruebas se configuró el modelo WRF.

10.5.1 Pruebas de conexión

10.5.1.1 Prueba 1

Descripción

Se hicieron pruebas de conexión desde el WRF Runner, utilizando la librería *LaravelCollectiveRemote*. Además de la prueba de conexión, se realizó pruebas de ejecución de comandos y captura de la salida de los comandos.

Resultados

En esta prueba se observaron algunos errores, donde el WRF Runner perdía la conexión (timeout) con el servidor luego de la ejecución de los comandos.

10.5.1.2 Prueba 2

Descripción

Se procedió a examinar la documentación de la librería y se decidió aumentar la configuración del tiempo de conexión de 10 segundos (valor por defecto) a 60 segundos.

Resultados

Se observó que todas las conexiones y comandos enviados al servidor con el ajuste de configuración hecho se realizaron correctamente.

10.6 Validación de pronósticos del WRF

Como se mencionó en el punto 8.5, se identificaron las cinco variables a ser empleadas en la validación, y se determinó el período de tiempo para generar los pronósticos en base a los datos disponibles en la DMH.

El período definido es del día 3 de noviembre de 2016 al 3 de diciembre de 2016.

Los datos de entrada del modelo global GFS disponibles corresponden al ciclo de las 18 UTC y cada ejecución del modelo generará 72 horas de pronóstico.

10.6.1 Análisis de los archivos GFS

10.6.1.1 Prueba 1

Descripción

Se realizó un análisis de los archivos de salida del modelo global GFS disponibles en el Data Center de la DMH.

Resultados

Se hallaron datos disponibles para los meses de octubre, noviembre y diciembre del 2016. Al verificar consistencia de los datos se encontraron que los meses de octubre y diciembre carecían de algunos archivos. Afortunadamente, noviembre estaba completo por lo que se resolvió utilizar el período del 3 de noviembre de 2016 al 3 de diciembre de 2016, completando de esta forma, 31 días.

10.6.2 Ejecución de las Simulaciones

10.6.2.1 Prueba 1

Descripción

Se programaron las simulaciones para los 31 días resultantes de la prueba anterior. Tres sitios fueron definidos como referenciales, de los cuales se extraerán las variables definidas en el punto 8.5, estos son: el aeropuerto de Encarnación, la Facultad de Agronomía de Hohenau y la base aérea de Capitán Meza. Todos estos tienen la misma latitud y longitud que las EMAS ubicadas en estos puntos, lo que permitió comparar los datos observados con lo pronosticado por el modelo.

Considerando los tiempos de proceso en el clúster de RPi's (aproximadamente 279 horas) detallados en el punto 10.4.1.7, se decidió realizar las simulaciones en el servidor de pruebas proveído por la DMH.

Resultados

Las simulaciones se efectuaron correctamente para los 30 días programados, demorando un total de 19 horas de proceso en el servidor de pruebas. En la Tabla 12 se puede apreciar los tiempos de ejecución.

Fecha y ciclo de simulación	Hora Inicio	Hora Fin	Tiempo de ejecución
2016-11-03 18UTC	05:06:08	05:41:10	00:35:02
2016-11-04 18UTC	05:43:08	06:17:51	00:34:43
2016-11-05 18UTC	06:19:07	06:55:48	00:36:41
2016-11-06 18UTC	06:57:07	07:35:15	00:38:08
2016-11-07 18UTC	07:37:08	08:16:05	00:38:57
2016-11-08 18UTC	03:17:19	03:54:35	00:37:16
2016-11-09 18UTC	04:00:09	04:35:22	00:35:13
2016-11-10 18UTC	04:37:07	05:13:36	00:36:29
2016-11-11 18UTC	05:15:08	05:51:08	00:36:00
2016-11-12 18UTC	16:25:07	16:59:54	00:34:47
2016-11-13 18UTC	14:22:21	15:00:19	00:37:58
2016-11-14 18UTC	15:02:07	15:38:13	00:36:06
2016-11-15 18UTC	15:41:07	16:23:38	00:42:31

Fecha y ciclo de simulación	Hora Inicio	Hora Fin	Tiempo de ejecución
2016-11-16 18UTC	17:02:24	17:36:06	00:33:42
2016-11-17 18UTC	17:37:09	18:12:24	00:35:15
2016-11-18 18UTC	18:14:10	18:49:32	00:35:22
2016-11-19 18UTC	18:51:07	19:27:00	00:35:53
2016-11-20 18UTC	20:19:10	20:52:42	00:33:32
2016-11-21 18UTC	20:54:08	21:26:44	00:32:36
2016-11-22 18UTC	21:28:09	22:01:11	00:33:02
2016-11-23 18UTC	22:03:09	22:37:15	00:34:06
2016-11-24 18UTC	23:43:09	00:22:02	00:38:53
2016-11-25 18UTC	00:23:10	01:03:04	00:39:54
2016-11-26 18UTC	01:04:09	01:44:02	00:39:53
2016-11-27 18UTC	01:45:08	02:23:31	00:38:23
2016-11-28 18UTC	02:25:08	03:01:36	00:36:28
2016-11-29 18UTC	03:03:08	03:38:09	00:35:01
2016-11-30 18UTC	03:40:08	04:18:06	00:37:58
2016-12-01 18UTC	15:03:32	15:42:31	00:38:59
2016-12-02 18UTC	15:44:08	16:20:15	00:36:07
2016-12-03 18UTC	16:22:07	16:56:05	00:33:58

Tabla 12: Tiempos de ejecución del Modelo WRF en el Servidor de Pruebas

10.6.2.2 Análisis de los datos de las EMAS

10.6.2.3 Prueba 1

Descripción

Con los pronósticos generados para los tres sitios configurados, se realizó un análisis de los datos observados por las EMAS del aeropuerto de Encarnación, de la Facultad de Agronomía de Hohenau y de la base aérea de Capitán Meza.

Resultados

Se observó que la estación EMA del aeropuerto de Encarnación no posee datos de precipitación para el período comprendido entre el 3/11/2016 al 3/12/2016. Se consultó sobre la falta de datos con el Departamento de Sensores Remotos de la DMH y confirmaron que el pluviómetro de la estación no se encuentra en funcionamiento. Se decidió no utilizar los datos de esta estación.

Con relación a la Estación EMA de la Facultad de Agronomía de Hohenau, se observó que los datos de temperatura, presión atmosférica y dirección del viento, presentan lapsos de tiempo de los que no se tienen datos. Se decidió no utilizar los datos de esta estación.

La EMA de la base aérea de Capitán Meza presenta error en una sola lectura para todo el período, por lo que se decidió utilizar los datos de esta estación para hacer las comparaciones. Es la única de las tres estaciones analizadas que posee datos completos.

10.6.3 Comparación de los datos

10.6.3.1 Prueba 1

Descripción

Se procesaron los datos de la estación de la base aérea de Capitán Meza para realizar la comparación con el pronóstico generado por el modelo.

10.6.4 Resultados

Se realizó la comparación de los modelos y se constató que el modelo se ajusta a los datos observados. Se presentaron los datos comparados a los funcionarios de la DMH para su análisis y validación. En el punto 12.4 se presenta la nota donde se validan las salidas generadas por el modelo.

En las ilustraciones presentadas en las páginas siguientes se aprecian las comparaciones de los datos para las cinco variables de estudio (ver punto 8.5).

Temperatura a 2 metros.

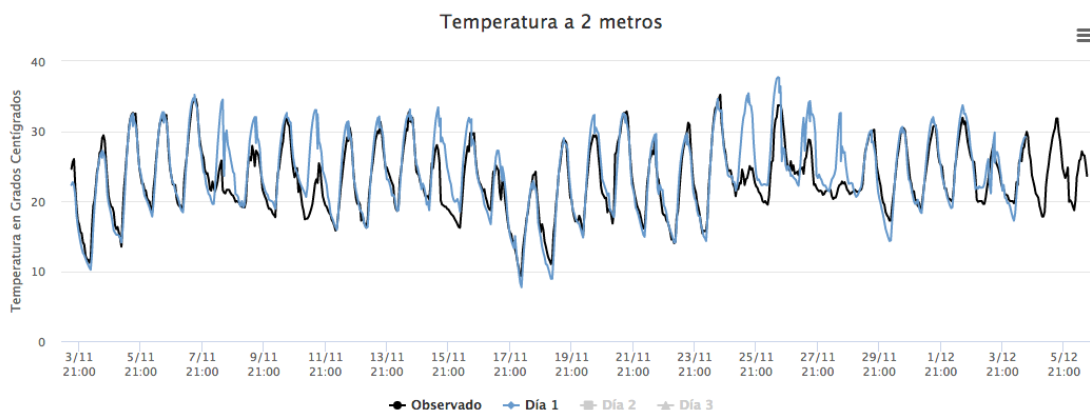


Ilustración 43: Validación del Modelo WRF - Temperatura - Día 1.

Fuente: Elaboración propia.

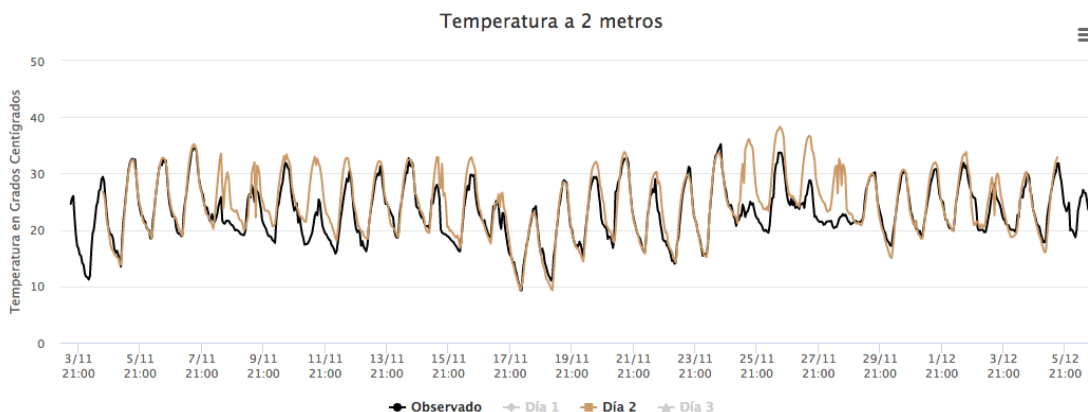


Ilustración 44: Validación del Modelo WRF - Temperatura - Día 2.

Fuente: Elaboración propia.

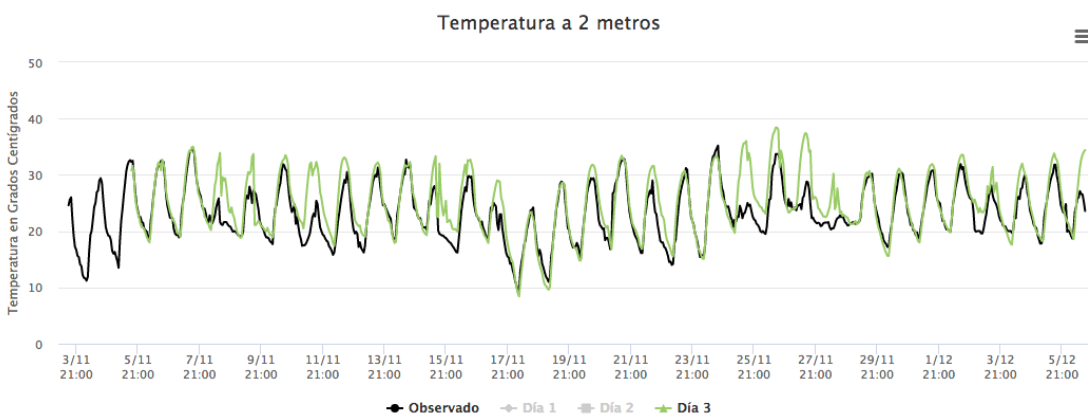


Ilustración 45: Validación del Modelo WRF - Temperatura - Día 3.

Fuente: Elaboración propia.

Presión atmosférica

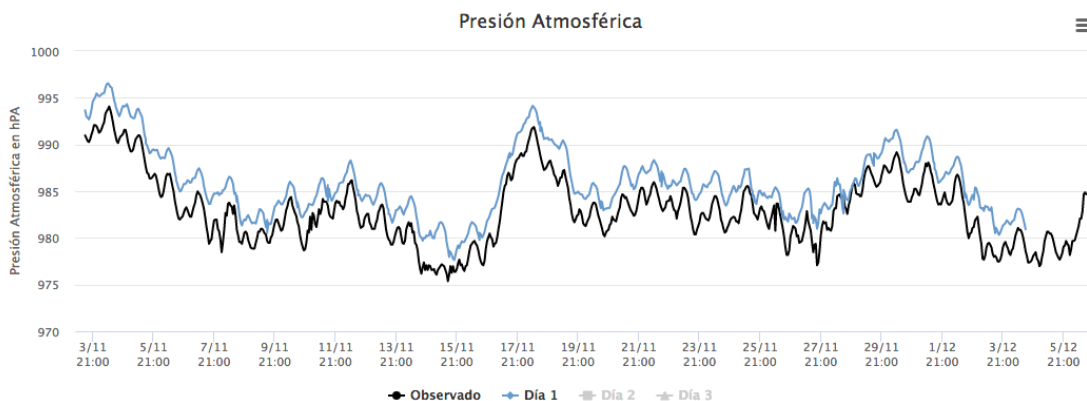


Ilustración 46: Validación del Modelo WRF - Presión Atmosférica - Día 1.
Fuente: Elaboración propia.

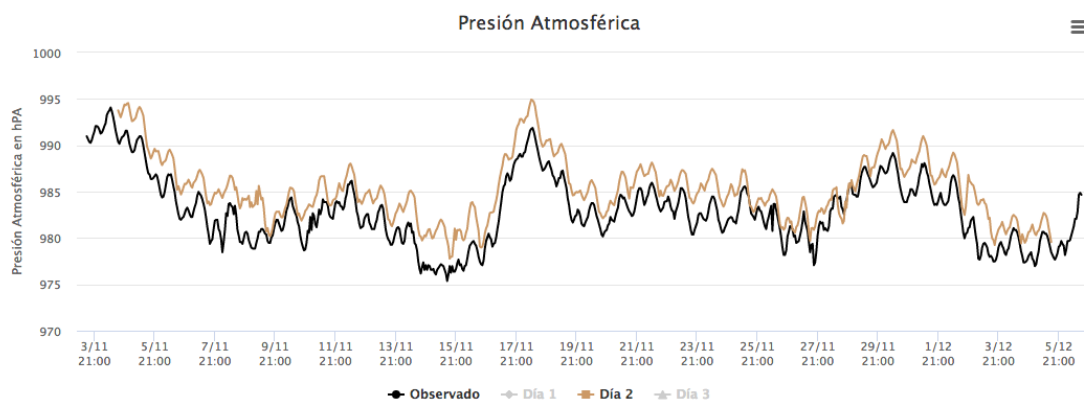


Ilustración 47: Validación del Modelo WRF - Presión Atmosférica - Día 2.
Fuente: Elaboración propia.

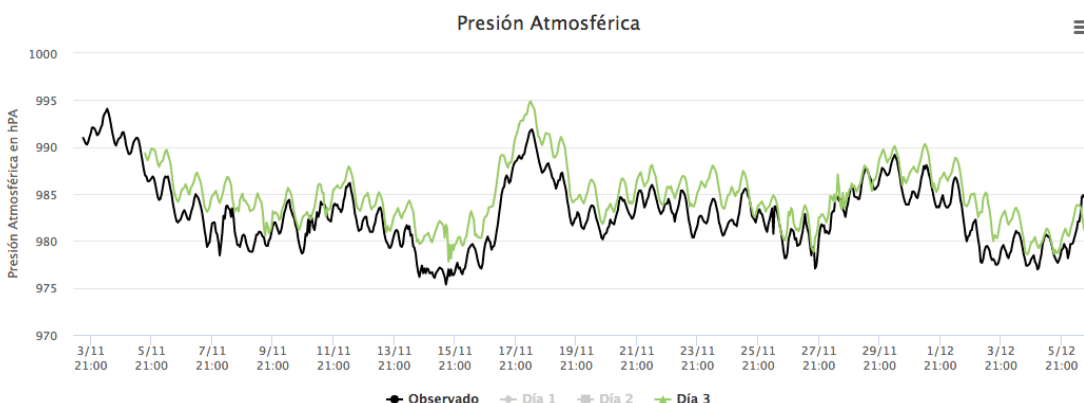


Ilustración 48: Validación del Modelo WRF - Presión Atmosférica - Día 3.
Fuente: Elaboración propia.

Dirección del viento

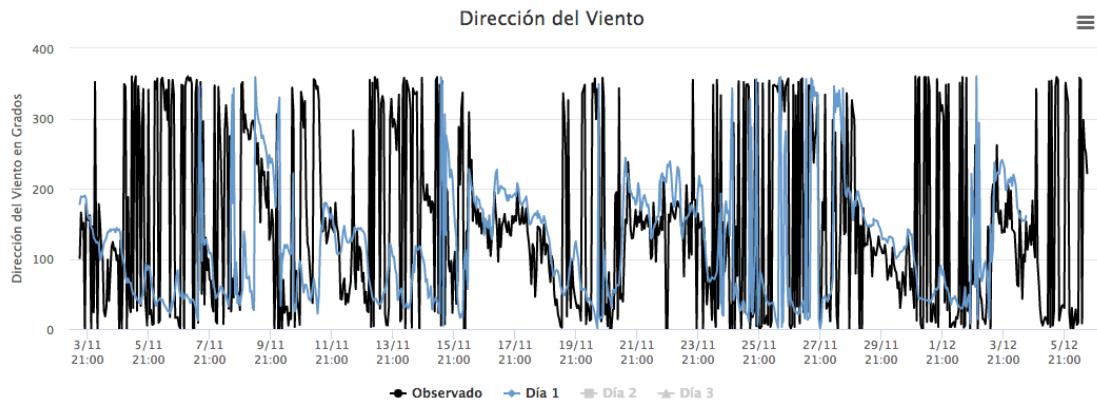


Ilustración 49: Validación del Modelo WRF - Dirección del Viento - Día 1.
Fuente: Elaboración propia.

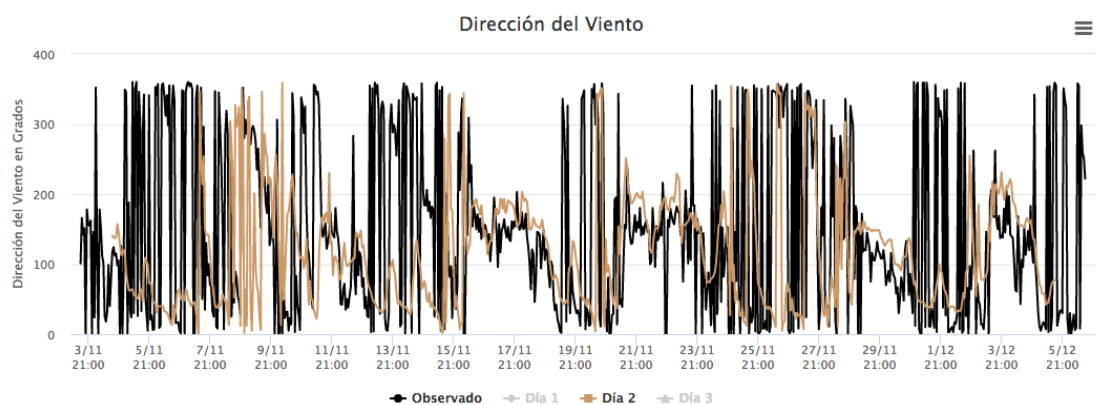


Ilustración 50: Validación del Modelo WRF - Dirección del Viento - Día 2.
Fuente: Elaboración propia.

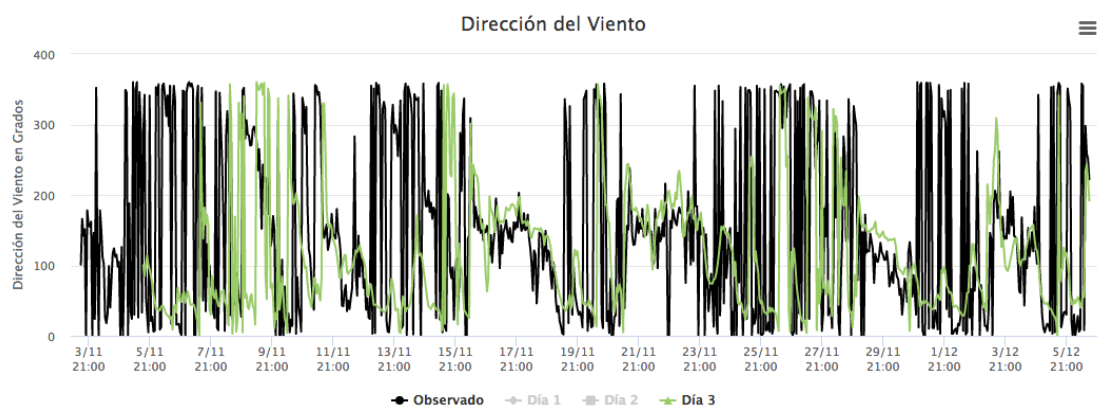


Ilustración 51: Validación del Modelo WRF - Dirección del Viento - Día 3.
Fuente: Elaboración propia.

Velocidad del viento

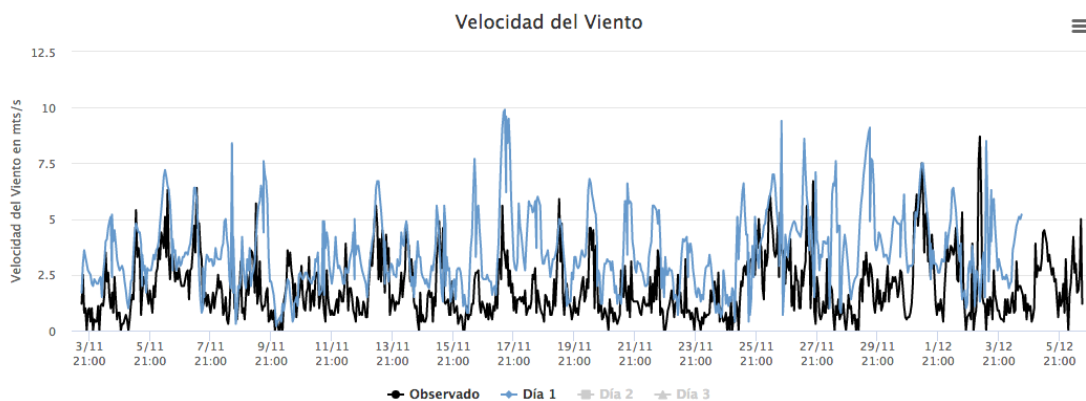


Ilustración 52: Validación del Modelo WRF - Velocidad del viento - Día 1.
Fuente: Elaboración propia.

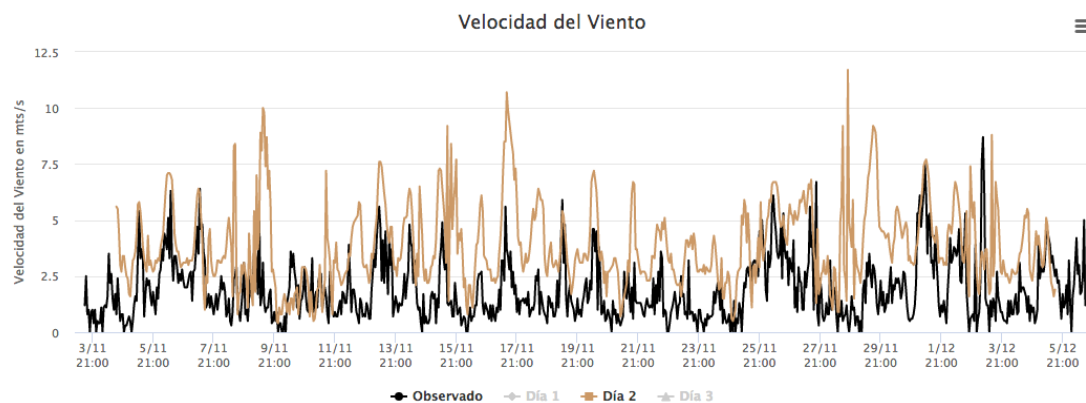


Ilustración 53: Validación del Modelo WRF - Velocidad del viento - Día 2.
Fuente: Elaboración propia.

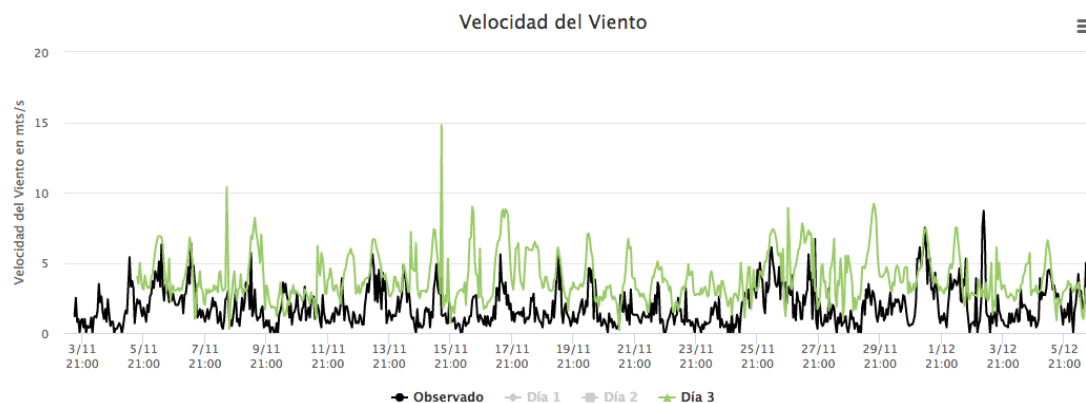


Ilustración 54: Validación del Modelo WRF - Velocidad del viento - Día 3.
Fuente: Elaboración propia.

Precipitación

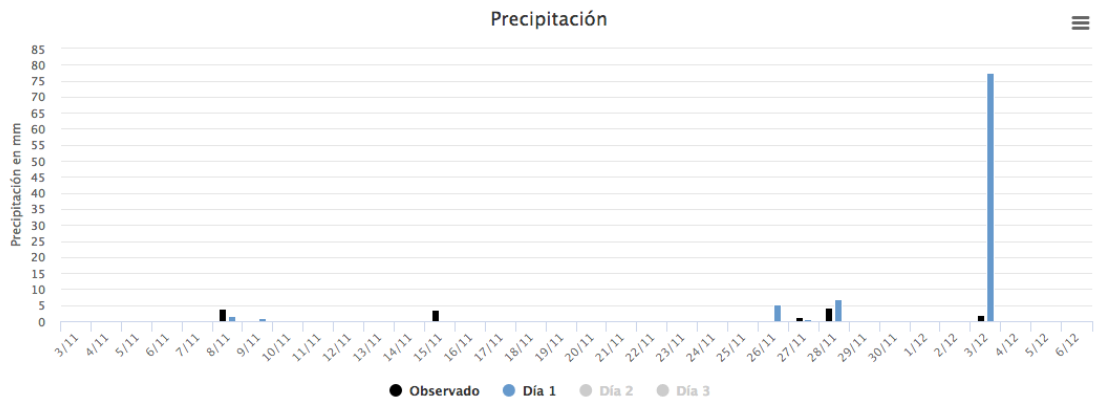


Ilustración 55: Validación del Modelo WRF - Precipitación - Día 1.
Fuente: Elaboración propia.

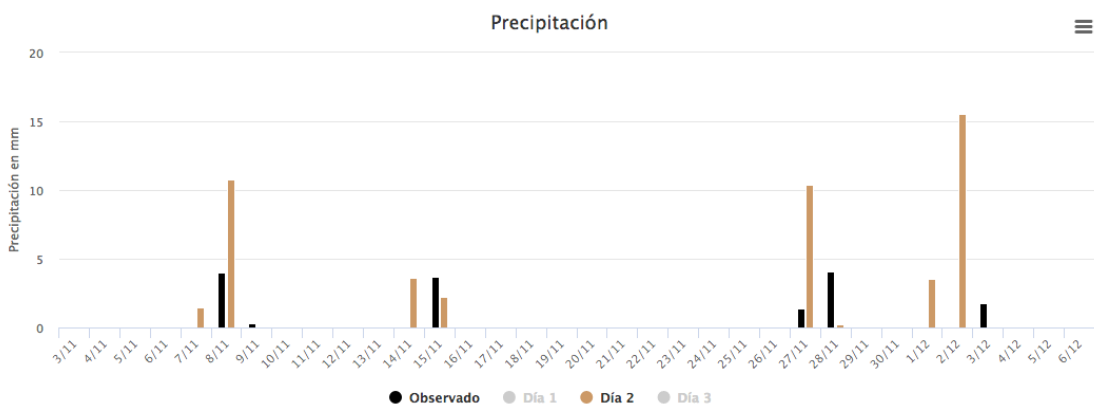


Ilustración 56: Validación del Modelo WRF - Precipitación - Día 2.
Fuente: Elaboración propia.

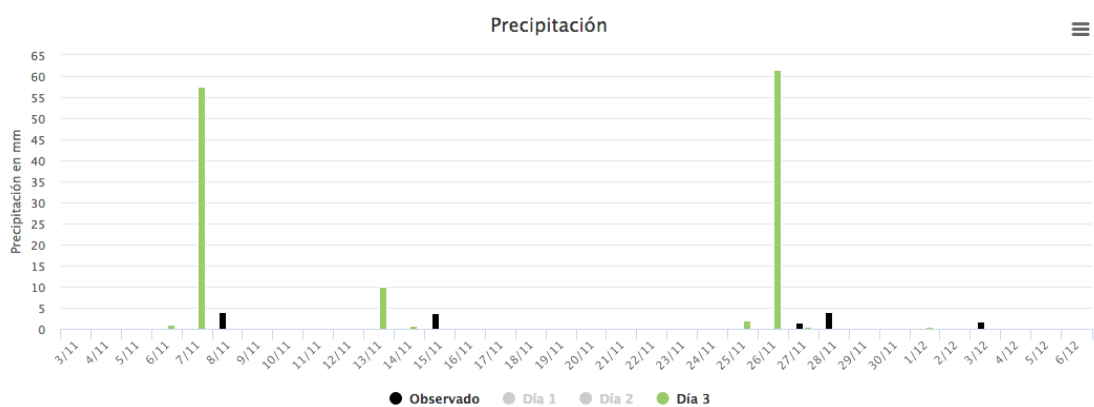


Ilustración 57: Validación del Modelo WRF - Precipitación - Día 3.
Fuente: Elaboración propia.

11 Conclusiones y trabajos futuros

11.1 Conclusiones

Tras las investigaciones realizadas durante el desarrollo de este trabajo de investigación, se puede afirmar que se ha cumplido con el objetivo general de *implementar el Modelo de Investigación y Pronóstico del Tiempo en un Clúster de Raspberry Pi's para la previsión meteorológica del tiempo en la ciudad de Encarnación*.

En el punto 6.5 se diseñó e implementó el Clúster cumpliendo el objetivo específico de *Implementar el Clúster de Raspberry Pi's*.

En los puntos 4.4 y 4.5 se determinaron los datos de entrada necesarios para el WRF, como así también, los procesos necesarios de pre procesamiento de datos, dando cumplimiento a los objetivos específicos *Determinar y recopilar datos para alimentar al modelo* y *Analizar datos y ajustarlos al modelo de entrada de datos del WRF*.

En los puntos 8.2, 8.3 y 10.4 se establecen los parámetros requeridos para la zona de estudio, cumpliendo de esta forma con el objetivo específico *Determinar los parámetros de adaptabilidad del modelo WRF a la zona de estudio*.

Los objetivos específicos *Analizar y comparar los datos devueltos por el modelo WRF*, *Contrastar datos obtenidos con observaciones de campo*, *Determinar necesidades de iteración en base a los datos comparados* y *Graficar resultados utilizando herramientas Open Source existentes* son abarcados en el capítulo 10.6.3

En el punto 12.4 del anexo se puede observar que la Dirección de Meteorología e Hidrología valida los pronósticos generados por el modelo, dando cumplimiento al objetivo específico de *Validar resultados utilizando estándares establecidos*.

En el punto 9.2, se relata el desarrollo del sistema que almacena los datos generados por el modelo, cumpliendo de esta forma con el objetivo específico *Poner en funcionamiento un servidor de datos generados por el modelo*.

Finalmente, en el Capítulo 9.3, se desarrolló una aplicación móvil para visualizar los resultados, cumpliendo el último objetivo específico *Desarrollar una aplicación móvil para visualizar los resultados*.

11.2 Trabajos futuros

Al finalizar este trabajo de investigación, se ha podido identificar varias líneas de investigación para futuros trabajos:

- Clúster
 - Implementación de un clúster similar utilizando otros tipos de computadoras de placa reducida, que cuenten con más recursos computacionales.
- Modelo WRF
 - Asimilación de datos en el modelo WRF utilizando OBSPROC con datos de globos sonda.
 - Asimilación de datos en el Modelo WRF utilizando WRFDA para datos de radar.
 - Generación de nuevos productos en base a los pronósticos generados para usos específicos como la agricultura, ganadería, fumigación aérea, etc.
- Sistema GEMHA
 - Aplicación de controles de calidad en base a reglas de validación en tiempo real.
 - Interpolación de datos no existentes con estaciones cercanas.

12 Anexo

12.1 Red de estaciones de la DMH

12.1.1 Estaciones Meteorológicas Automáticas

N°	Nombre Estación	Tipo	Tipo de Archivo	Latitud	Longitud
1	AGCDE	OTT	MIS	-25.519	-54.612
2	AIG	OTT	MIS	-25.456	-54.839
3	AISP	OTT	MIS	-25.236	-57.515
4	Arazape	DAVIS	SQLITE	-26.593	-56.852
5	Ayolas	DAVIS	SQLITE/ARC	-27.310	-56.940
6	Benjamín	OTT	MIS/No transmite	-24.963	-57.550
7	Caacupé	DAVIS	SQLITE/ARC	-25.390	-57.188
8	Caazapá	OTT	MIS	-26.170	-56.360
9	Capitán Meza	OTT	MIS	-26.834	-55.332
10	Cerrito	DAVIS	ARC	-27.345	-57.641
11	CFDO	DAVIS	SQLITE	-25.335	-57.340
12	Choré	DAVIS	SQLITE	-24.585	-56.257
13	CITEC	DAVIS	Sin datos/No transmite		
14	CMN	OTT	MIS	-25.286	-57.655
15	Concepción	OTT	MIS	-23.441	-57.430
16	San Cosme	DAVIS	SQLITE/ARC	-27.305	-56.186
17	Costanera	DAVIS	SQLITE/ARC	-25.277	-57.638
18	Díaz	Sin Datos	Sin datos/No transmite		
19	Emboscada	DAVIS	Sin datos/No transmite		
20	Encarnación	OTT	MIS	-27.306	-55.887
21	FCAH	DAVIS	SQLITE/ARC	-27.062	-55.642
22	FECOPROD	Históricos	ARC		
23	FIUNA	DAVIS	ARC	-25.331	
24	ITABO	DAVIS	ARC	-25.132	-54.512
25	Oviedo	OTT	MIS	-25.515	-56.408
26	PaiPukú	DAVIS	SQLITE/ARC	-22.804	-59.597
27	Pilar	OTT	MIS	-26.881	-58.318
28	Pedro J. Caballero	OTT	MIS	-22.640	-55.833
29	Politécnica	Sin datos	Sin datos/No transmite		
30	Pozo Colorado	OTT	MIS	-23.498	-58.790
31	Sajonia	DAVIS	ARC	-25.286	-57.655

Nº	Nombre Estación	Tipo	Tipo de Archivo	Latitud	Longitud
32	Salto	OTT	MIS	-24.032	-54.353
33	San Bernardino	OTT	MIS	-25.313	-57.300
34	Santaní	OTT	MIS	-24.665	-56.457
35	SJB	OTT	MIS	-26.671	-57.134
36	SND	DAVIS	SQLITE/ARC	-25.235	-57.314
37	UCA	DAVIS	ARC	-25.474	-57.570
38	Vallemí	OTT	MIS	-22.160	-57.943
39	Villarrica	OTT	MIS	-25.754	-56.438
40	Villeta	DAVIS	Sin datos/No transmite		
41	Yvytu	DAVIS	Sin datos/No transmite		

Tabla 13: Listado de estaciones automáticas. Red DMH

12.1.2 Estaciones Hidrológicas Automáticas

Nº	Nombre Estación	Tipo	Tipo de Archivo	Latitud	Longitud
1	Pozo Loma Pyta	OTT	MIS		
2	Arroyo San Lorenzo – Capiata	OTT	MIS	-2,531,558	-57,454,997
3	Rio Yhaguy – Arroyos y Esteros	OTT	MIS	-22.398725	-56.416569
4	Pozo Santa Librada – Itá	OTT	MIS	-26.312325	-54.416569
5	Capiibary – Jesús y Trinidad	OTT	MIS	-25.243413	-56.456934
6	Pozo – Capiata	OTT	MIS		
7	Rio Tebicuary – Yuty	OTT	MIS	-2,531,558	-57,454,997
8	Capiibary – Hohenau	OTT	MIS	-22.398725	-56.416569
9	Pozo – Ñemby	OTT	MIS	-26.312325	-54.416569
10	Rio Tebicuary – Villa Florida	OTT	MIS	-25.243413	-56.456934
11	Arroyo Yukyrymi – Areguá	OTT	MIS	-22.1256	-54.329867
12	Rio Tebicuarymi – Iturbe	OTT	MIS	-25.347725	-57.416569

Tabla 14: Datos en bruto EHAS - DMH

12.1.3 Datos en bruto de a Setiembre 2016

Carpeta Estación	Espacio Utilizado (MB)	Montado en
ftp/ehga	142	/mnt/ftp/ehga
ftp/ehms	0.32	/mnt/ftp/ehms
ftp/ehms/nivel	280	/mnt/ftp/ehms/nivel
ftp/emas/agcde	134	/mnt/ftp/emas/agcde
ftp/emas/aig	140	/mnt/ftp/emas/aig
ftp/emas/aisp	380	/mnt/ftp/emas/aisp
ftp/emas/arazape	70	/mnt/ftp/emas/arazape
ftp/emas/arduino	2	/mnt/ftp/emas/arduino
ftp/emas/ayolas	37	/mnt/ftp/emas/ayolas

**Previsión meteorológica del tiempo para Encarnación en un clúster de Raspberry Pi's
utilizando el Modelo de Investigación y Pronóstico del Tiempo (WRF)**

Carpeta Estación	Espacio Utilizado (MB)	Montado en
ftp/emas/benjamin	77	/mnt/ftp/emas/benjamin
ftp/emas/caacupe	68	/mnt/ftp/emas/caacupe
ftp/emas/caazapa	165	/mnt/ftp/emas/caazapa
ftp/emas/capmeza	329	/mnt/ftp/emas/capmeza
ftp/emas/cartas	28	/mnt/ftp/emas/cartas
ftp/emas/cerrito	27	/mnt/ftp/emas/cerrito
ftp/emas/cfdo	11	/mnt/ftp/emas/cfdo
ftp/emas/chore	26	/mnt/ftp/emas/chore
ftp/emas/citec	2.2	/mnt/ftp/emas/citec
ftp/emas/cmn	355	/mnt/ftp/emas/cmn
ftp/emas/concepcion	337	/mnt/ftp/emas/concepcion
ftp/emas/cosme	32	/mnt/ftp/emas/cosme
ftp/emas/costanera	16	/mnt/ftp/emas/costanera
ftp/emas/datosjson	475	/mnt/ftp/emas/datosjson
ftp/emas/diaz	0.2	/mnt/ftp/emas/diaz
ftp/emas/emboscada	4.1	/mnt/ftp/emas/emboscada
ftp/emas/encarnacion	132	/mnt/ftp/emas/encarnacion
ftp/emas/especificaciones	50	/mnt/ftp/emas/especificaciones
ftp/emas/estaciones	812	/mnt/ftp/emas/estaciones
ftp/emas/fcah	43	/mnt/ftp/emas/fcah
ftp/emas/fcaydr	0.558	/mnt/ftp/emas/fcaydr
ftp/emas/fecoprod	183	/mnt/ftp/emas/fecoprod
ftp/emas/fiuna	5.4	/mnt/ftp/emas/fiuna
ftp/emas/itabo	1.9	/mnt/ftp/emas/itabo
ftp/emas/misbruto	72	/mnt/ftp/emas/misbruto
ftp/emas/oviedo	259	/mnt/ftp/emas/oviedo
ftp/emas/paipuku	86	/mnt/ftp/emas/paipuku
ftp/emas/pilar	126	/mnt/ftp/emas/pilar
ftp/emas/pjc	16	/mnt/ftp/emas/pjc
ftp/emas/politecnica	0.383	/mnt/ftp/emas/politecnica
ftp/emas/pozo	213	/mnt/ftp/emas/pozo
ftp/emas/pruebas	208	/mnt/ftp/emas/pruebas
ftp/emas/pti	21	/mnt/ftp/emas/pti
ftp/emas/sajonia	1.9	/mnt/ftp/emas/sajonia
ftp/emas/salto	249	/mnt/ftp/emas/salto
ftp/emas/samber	334	/mnt/ftp/emas/samber
ftp/emas/santani	165	/mnt/ftp/emas/santani
ftp/emas/sjb	232	/mnt/ftp/emas/sjb
ftp/emas/snd	53	/mnt/ftp/emas/snd
ftp/emas/sondeo	3.3	/mnt/ftp/emas/sondeo
ftp/emas/src	0.33	/mnt/ftp/emas/src
ftp/emas/uca	0.673	/mnt/ftp/emas/uca
ftp/emas/vallemi	167	/mnt/ftp/emas/vallemi
ftp/emas/villarrica	188	/mnt/ftp/emas/villarrica

Carpeta Estación	Espacio Utilizado (MB)	Montado en
ftp/emas/villeta	7.4	/mnt/ftp/emas/villeta
ftp/emas/yvytu	4.3	/mnt/ftp/emas/yvytu
bd/emas	39	/mnt/bd/emas
bd/fiuna	34	/mnt/bd/fiuna

Tabla 15: Datos en bruto EMAS - DMH

12.2 Costos del proyecto

Los gastos emergentes de la elaboración del proyecto han sido absorbidos por los tesistas.

Descripción	Costo US\$.
Adquisición de Raspberry Pi's	1.150
Adquisición de Switch para el clúster	100
Montaje de clúster de RPi's	350
Libros, revistas y suscripciones	48
Viáticos	500
Otros materiales – servicios	95
Plantillas – licencias	12
Servidor de pruebas	0
Servidor virtual del sistema GEMHA	0
Costo total del proyecto	2.255

Tabla 16: Costos del proyecto

12.3 Montaje del clúster de RPi's



Ilustración 58: Clúster RPi's - Mediciones. Fuente: Elaboración propia.

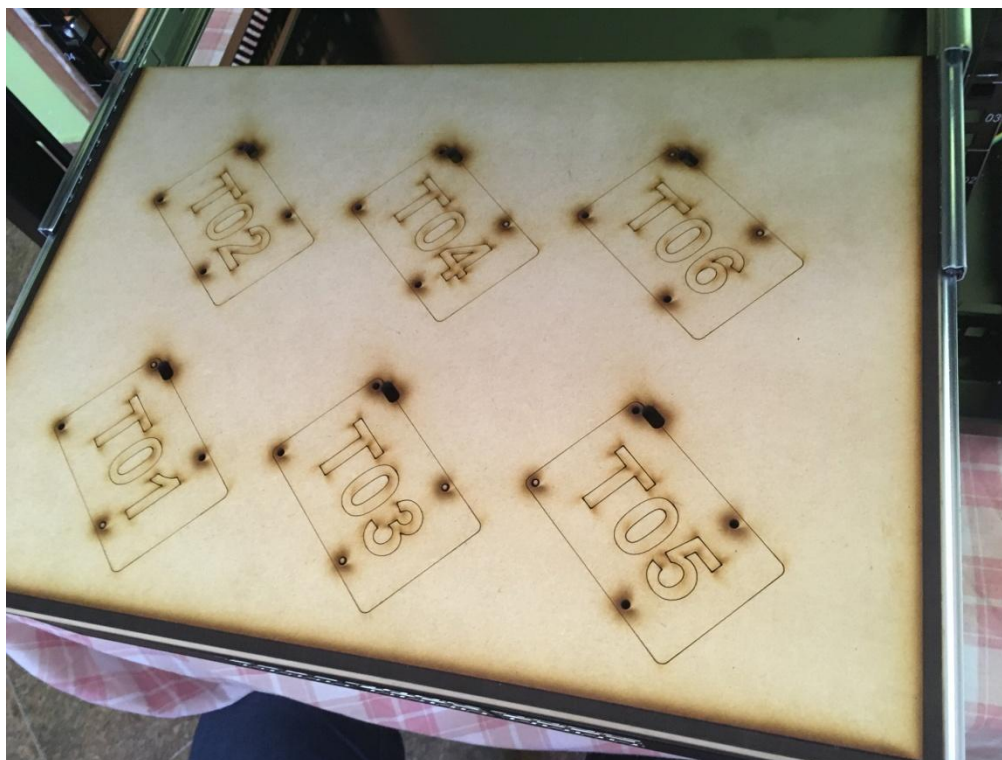


Ilustración 59: Clúster RPi's - Corte láser MDF. Fuente: Elaboración propia.

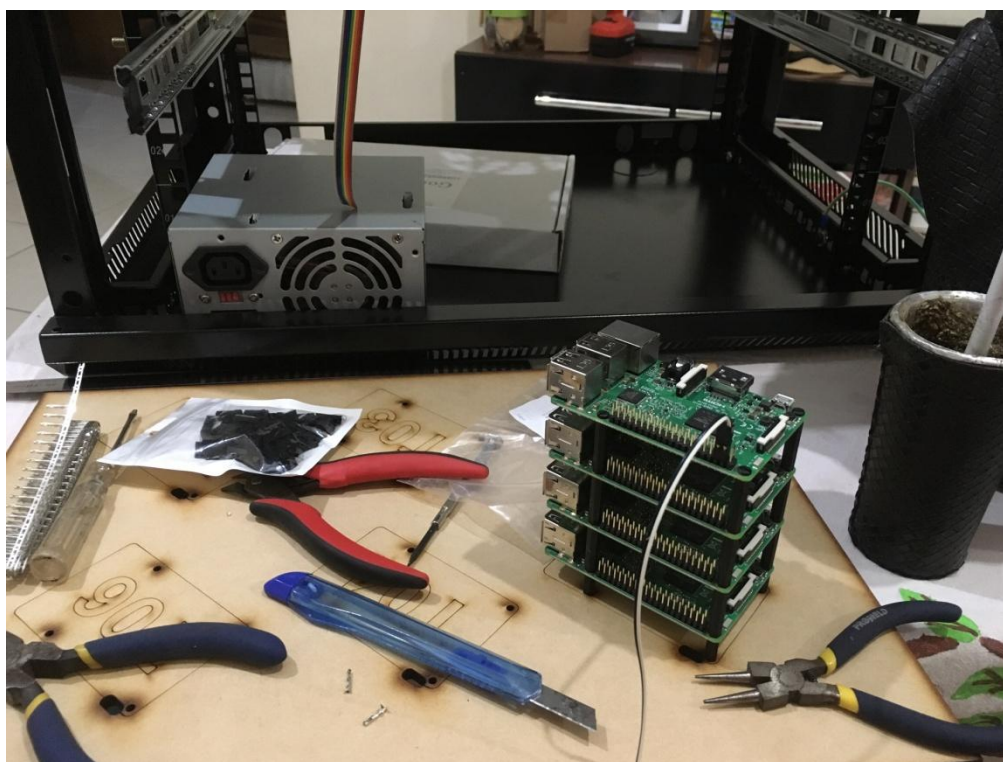


Ilustración 60: Clúster RPi's - Verificación de medidas y cableado. Fuente: Elaboración propia.

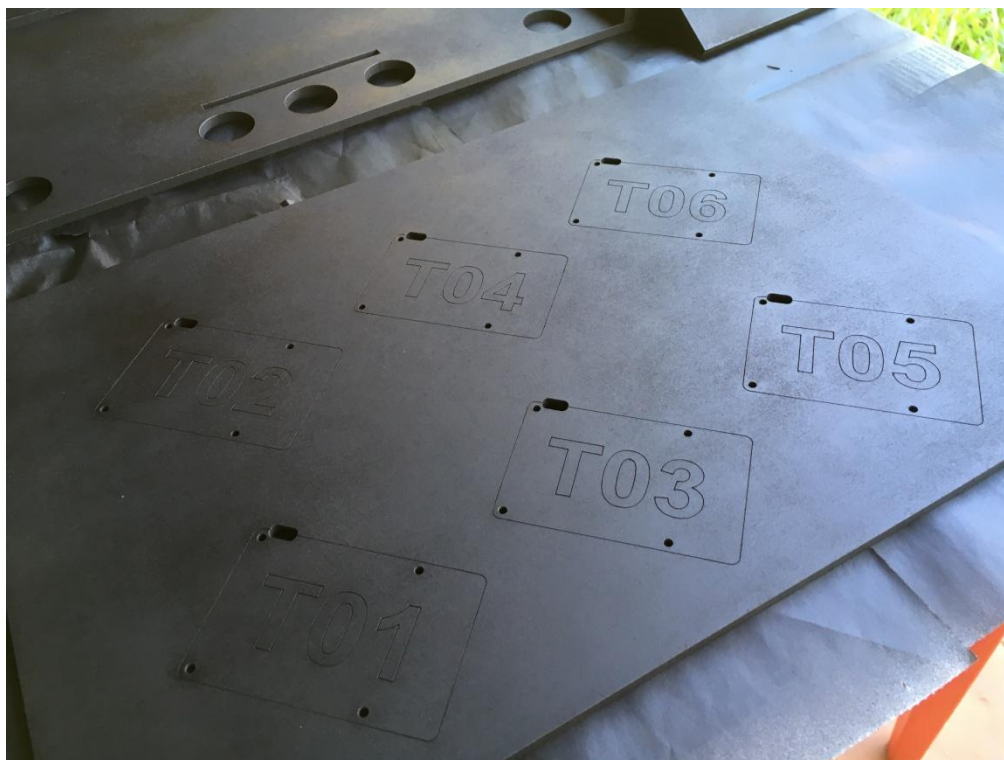


Ilustración 61: Clúster RPi's - Pintura. Fuente: Elaboración propia.



Ilustración 62: Clúster RPi's - Cableado de alimentación. Fuente: Elaboración propia.

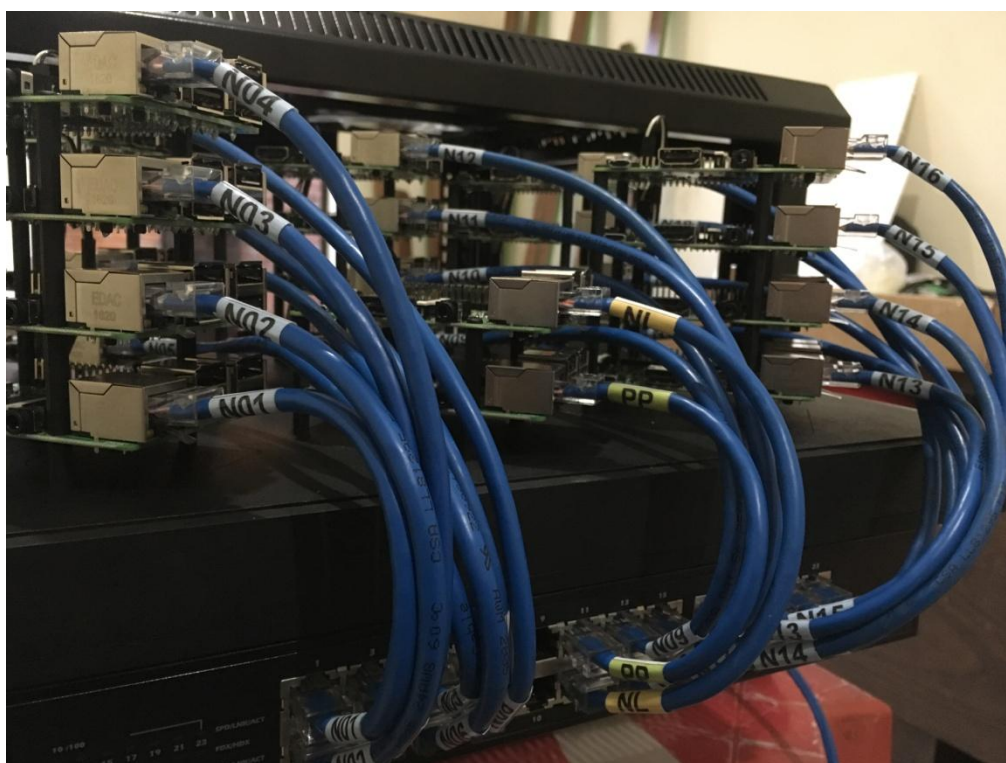


Ilustración 63: Clúster RPi's -Cableado de Red. Fuente: Elaboración propia.



Ilustración 64: Clúster RPi's - Funcionamiento. Fuente: Elaboración propia.

12.4 Carencia de un modelo de predicción numérica para Encarnación



Asunción, 08 de Marzo de 2017

Señores
Ever Barreto y Nicolás Rodríguez
Tesisistas
Universidad Católica de Itapúa
Ciudad

Por la presente, tengo el agrado de dirigirme a Ustedes, en respuesta a la nota remitida en fecha 15 de diciembre del 2016, y comunicarles que la Dirección de Meteorología e Hidrología no cuenta con un modelo de predicción numérica en ejecución, operacional, con un dominio de alta resolución para la Ciudad de Encarnación.

Sin otro particular, aprovecho la oportunidad para salidarlos muy cordialmente.




Lic. Fis. Fernando Pío Barrios
Gerente de Sistemas de Observaciones
Meteorológicas

Ilustración 65: Nota sobre carencia de un modelo de predicción numérica del tiempo para Encarnación.
Fuente: DMH – DINAC.

12.5 Validación del modelo WRF



Ilustración 66: Nota validación del modelo WRF. Fuente: DMH – DINAC.

12.6 Presentación Centro de monitoreo y alerta hidrometeorológica de Encarnación



Ilustración 67: Disertación MSc. Lic. Julián Báez. Fuente: Elaboración propia.



Ilustración 68: Disertación Ever Barreto y Marico Tamada. Fuente: Elaboración propia.

12.7 Taller de Hidrometeorología Aplicada



Ilustración 69: Taller Hidrometeorología Aplicada - Exposición Víctor Álvarez DMH.
Fuente:Elaboración propia.



Ilustración 70: Taller Hidrometeorología Aplicada – Asistentes.
Fuente:Elaboración propia.



Ilustración 71: Estaciones convencional y automática - Aeropuerto de Encarnación.
Fuente: Elaboración propia.



Ilustración 72: Observador de la DINAC presentando EMA – Aeropuerto de Encarnación.
Fuente: Elaboración propia.

12.8 Visita técnica a la DMH



Ilustración 73: Visita técnica a la DMH - Estación Automática OTT.
Fuente: Elaboración propia.



Ilustración 74: Visita técnica a la DMH - Datacenter.
Fuente: Elaboración propia.

12.9 Montaje de Rack con servidor de prueba en el Datacenter de la DMH



Ilustración 75: Servidor de prueba DMH. Fuente: Elaboración propia.



Ilustración 76: Servidor de prueba en nuevo Rack. Fuente: Elaboración propia.

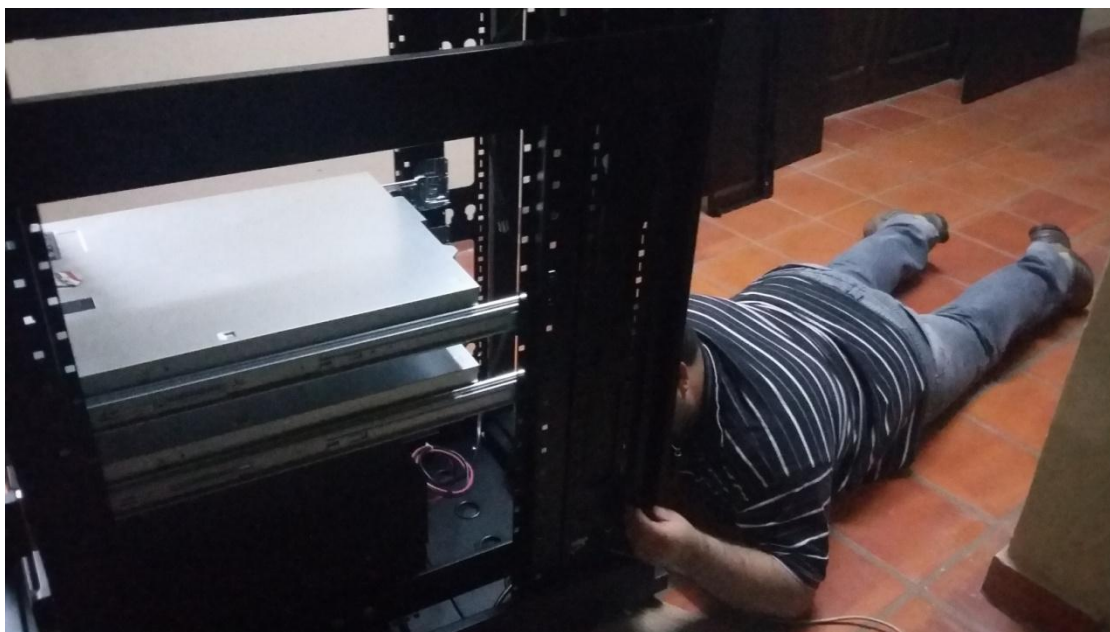


Ilustración 77: Nicolás Rodríguez instalando UPS y PDU. Fuente: Elaboración propia.



Ilustración 78: Ing. Wilson Caballero transportando el nuevo Rack. Fuente: Elaboración propia.



Ilustración 79: Nuevo Rack de la DMH instalado. Fuente: Elaboración propia.

13 Glosario de términos

API: Application Programming Interface.

Aplicaciones Web: codificado en lenguaje soportado por los navegadores.

APP: es una aplicación móvil diseñada para ser ejecutada en dispositivos móviles.

Arquitectura ARM: microprocesador que maneja un sistema de instrucciones realmente simple lo que le permite ejecutar tareas con un mínimo consumo de energía.

Base de Datos: conjunto de datos pertenecientes a un mismo contexto y almacenado sistemáticamente.

Compilación: es el proceso de traducción de programas fuente a programas objeto.

Demonios (daemons): programa que se ejecuta en segundo plano.

Framework: estructura conceptual y tecnológica de soporte definido.

Fuente: dispositivo que se encarga de transformar la corriente alterna de la línea eléctrica en corriente continua o directa.

GET: método HTTP.

HTML: es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. Las siglas corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.

HTTP: protocolo utilizado en cada transacción de la World Wide Web.

Interfaz: conexión física y funcional entre dos sistemas. Zona de comunicación o acción de un sistema sobre otro

JSON: formato ligero para el intercambio de datos.

Linux: sistema operativo libre basado en UNIX. Es considerado uno de los principales ejemplos del software libre y de código abierto en todo el mundo.

MVC: patrón de arquitectura de software, modelo, vista, controlador.

Nodo: punto de intersección o unión de varios elementos que confluyen en el mismo lugar. En redes de computadoras, cada una de las máquinas es un nodo.

Open Source: expresión con la que se conoce al software distribuido y desarrollado libremente.

Parametrización: es la organización y estandarización de la información que se ingresa en un sistema.

Plataforma: entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones.

Pronóstico: conocimiento anticipado de lo que sucederá en un futuro mediante la lectura de ciertas variables que se suceden.

Restful: transferencia de estados representacional.

Sensor: dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno.

Sitio Web: colección de páginas de internet relacionadas y comunes a un dominio.

Software: soporte lógico de un sistema informático.

Switch: es un dispositivo de interconexión de redes informáticas.

UNIX: sistema operativo portable, multitarea y multiusuario.

14 Bibliografía

- [1] Manolo Palomares Calderón de la Barca, *Breve Historia de la Meteorología*. España, 2015, pp. 1-7.
- [2] DMH - DINAC, Taller de Hidrometeorología Aplicada - UCI, 2016.
- [3] Peter Lynch, "The origins of computer weather prediction and climate modeling," *Journal of Computational Physics*, vol. 227, no. 7, Mar. 2008.
- [4] Peter Lynch, "Weather Prediction by Numerical Process," *Cambridge University Press*, 2006.
- [5] J. G. Charney, R. Fjörtoft, and J. Von Neuman, "Numerical Integration of the Barotropic Vorticity Equation," vol. 2, no. 4, 1950.
- [6] Kristine Harper, Louis W. Uccellini, Eugenia Kalnay, Kenneth Carey, and Lauren Morone, "50th Anniversary of Operational Numerical Weather Prediction," vol. 88, no. 5, 2007.
- [7] Norman A. Phillips, "The general circulation of the atmosphere: A numerical experiment," vol. 82, no. 352, 1956.
- [8] L.M. Leslie and G.S Dietachmeyer, "Real-time limited area numerical weather prediction in Australia: a historical perspective," Dec. 1992.
- [9] David J. Stensrud, "Parameterization schemes: keys to understanding numerical weather prediction models," 2007.
- [10] T. N. Krishnamurti, "Numerical weather prediction," vol. 27, 1995.

- [11] Bradley A. Ballish and V. Krishna Kumar, "Investigation of systematic differences in aircraft and radiosonde temperatures with implications for nwp and climate studies , " *American Meteorological Society*, Nov. 2008.
- [12] National Data Buoy Center. (2009, Jan.) NOAA. [Online].
"http://www.vos.noaa.gov/vos_scheme.shtml"
- [13] V. S. Sunderam, G. Dick van Albada, M. A. Peter, and J. J. Dongarra Sloot, "Computational Science -- ICCS 2005," in *Computational Science -- ICCS 2005*. Atlanta, USA, 2005, p. 132.
- [14] Walter Zwiefelhofer, Norbert Kreitz, and European Centre for Medium Range Weather Forecasts,, 2001, p. 276.
- [15] Johnny C. L. & Jeffrey D. Kepert Chan, *Global Perspectives on Tropical Cyclones: From Science to Mitigation.*, 2010, pp. 295-296.
- [16] James R. Holton, "An introduction to dynamic meteorology, Volume 1," in *An introduction to dynamic meteorology, Volume 1.*, 2004, p. 480.
- [17] Molly E. Brown, "Famine early warning systems and remote sensing data," in *Famine early warning systems and remote sensing data.*, 2008, p. 121.
- [18] Masami & Shiro Ohmori Narita, "Improving precipitation forecasts by the operational Nonhydrostatic Mesoscale Model with the Kain-Fritsch Convective Parameterization And Cloud Microphysics," Aug. 2007.
- [19] David J. Stensrud, "Parameterization schemes: keys to understanding numerical weather prediction models," in *Parameterization schemes: keys to understanding numerical weather prediction models.*: Cambridge University Press, 2007, pp. 6, 12-14.
- [20] Thomas Tomkins Warner, "Numerical Weather and Climate Prediction," p. 259, 2010.
- [21] WRF-NMM Users Page. [Online]. "http://www.dtcenter.org/wrf-nmm/users/"

- [22] EMC NOAA, "The Global Forecast System (GFS) - Global Spectral Model (GSM)," May 2016.
- [23] NCEP NOAA. [Online].
"http://www.nco.ncep.noaa.gov/pmb/nwprod/prodstat_new/"
- [24] "Weather lessons from Superstorm Sandy," *USA Today*, Oct. 2013.
- [25] "The European forecast model already kicking America's butt just improved," *Ars Technica*, Nov. 2016.
- [26] "Supercomputer quietly puts U.S. weather resources back on top," *USA Today*, Feb. 2016.
- [27] "National Weather Service will boost its supercomputing capacity tenfold," *Ars Technica*, Jan. 2015.
- [28] WMO/OMM/BMO, *Vocabulario Meteorológico Internacional (OMM-N° 182)*., 1992.
- [29] OMM, *Guía de prácticas climatológicas (OMM N° 100)*., 2011.
- [30] The PHP Group, *PHP Manual*., 2017.
- [31] Python Software Foundation. [Online]. "https://docs.python.org/3/license.html"
- [32] University Corporation for Atmospheric Research , *NCL User Guide*., 2017.
- [33] PostgreSQL. postgresql.org. [Online].
"http://www.postgresql.org/es/sobre_postgresql"
- [34] Keith Rarick. beanstalkd. [Online]. "http://kr.github.io/beanstalkd/"
- [35] Ionic Framework. [Online].
"http://ionicframework.com/docs/guide/preface.html"

- [36] University Corporation for Atmospheric Research ,
"Compilation Tutorial," 2010.
- [37] *WRF-ARW V3: User's Guide.*, 2017, ch. 1, p. 1.
- [38] Simon Cox, "Steps to make Raspberry Pi Supercomputer," Sep. 2012.
- [39] *WRF-ARW V3: User's Guide.*, 2017, ch. 5, p. 66.
- [40] *WRF-ARW V3: User's Guide.*, 2017, ch. 5, p. 35.



**Universidad Católica
“Nuestra Señora de la Asunción”
Campus Itapúa**

Facultad de Ciencias y Tecnología

Ingeniería Informática

*Previsión meteorológica del tiempo para Encarnación en un
clúster de Raspberry Pi's utilizando el Modelo de Investigación
y Pronóstico del Tiempo (WRF)*

Mesa Examinadora

1. _____
2. _____
3. _____
4. _____

Calificación: _____

Fecha: _____